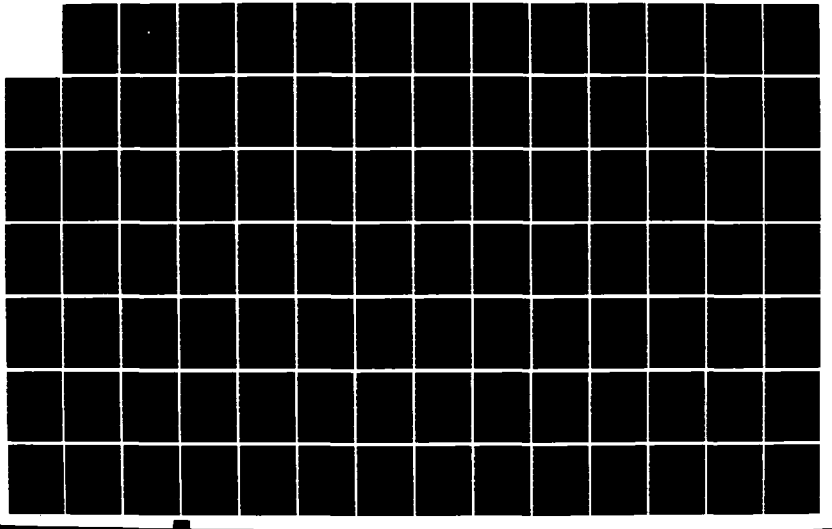
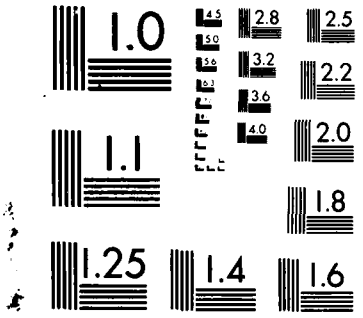
AD-A174 507 KNOT SELECTION FOR LEAST SQUARES APPROXIMATION USING THIN PLATE SPLINES(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA J R MCMAHON JUN 86 1/2

UNCLASSIFIED F/G 12/1 NL





(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

KNOT SELECTION FOR LEAST SQUARES
APPROXIMATION USING THIN PLATE SPLINES

John R. McMahon

June 1986

Thesis Advisor: Richard Franke

Approved for public release; distribution unlimited.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS A174 507
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (If applicable) 53	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11 TITLE (Include Security Classification)
KNOT SELECTION FOR LEAST SQUARES APPROXIMATION USING THIN PLATE SPLINES

12 PERSONAL AUTHOR(S)
McMahon, John R.

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Master's Thesis	FROM _____ TO _____	1986 June 10	98

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Knot Selection, Dirichlet Tesselation, Data Representation, Surface Fitting, Scattered Data, Thin Plate Spline, Least Squares Approximation, FORTRAN Subroutine

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Given a large set of scattered data (xi, yi, fi), a method for selecting a significantly smaller set of knot points which will represent the larger set is described, leading to a package of FORTRAN subroutines. The selection of the knot point locations is based on the minimization of the sum of the squares of the difference between the average number of points per Dirichlet tile and the actual number of points in each tile, subject to the constraint that each knot is located at the centroid of its tile. The pertinent theoretical and computational aspects of the subroutines are introduced and described in detail. Using the least squares thin plate spline approximation method for constructing surfaces, various test surfaces are examined and compared to surfaces obtained using smoothing splines and the bicubic Hermite approximation method. The FORTRAN subroutines are made available to prospective users through a point of contact.

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Richard Franke	22b TELEPHONE (Include Area Code) (408)646-2758 22c OFFICE SYMBOL Code 53Fe

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE

Approved for public release; distribution is unlimited.

Knot Selection for Least Squares Approximation Using Thin Plate Splines

by

John R. McMahon
Captain, United States Army
B.S., Syracuse University, 1977

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
June 1986

Author: John R. McMahon
John R. McMahon

Approved by: Richard Franke
Richard Franke, Thesis Advisor

Charles W. Schelin
Charles W. Schelin, Second Reader

Gordon Latta
Gordon Latta, Chairman,
Department of Mathematics

Kneale T. Marshall
Kneale T. Marshall,
Dean of Information and Policy Sciences

ABSTRACT

Given a large set of scattered data (x_i, y_i, f_i), a method for selecting a significantly smaller set of knot points which will represent the larger set is described, leading to a package of FORTRAN subroutines. The selection of the knot point locations is based on the minimization of the sum of the squares of the difference between the average number of points per Dirichlet tile and the actual number of points in each tile, subject to the constraint that each knot is located at the centroid of its tile. The pertinent theoretical and computational aspects of the subroutines are introduced and described in detail. Using the least squares thin plate spline approximation method for constructing surfaces, various test surfaces are examined and compared to surfaces obtained using smoothing splines and the bicubic Hermite approximation method. The FORTRAN subroutines are made available to prospective users through a point of contact.

A-1

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. GENERAL BACKGROUND AND ORIGIN OF THE PROBLEM

A. INTRODUCTION

The problem of fitting a surface to given data has been addressed in many different ways and several programs are currently available which enable one to deal with the problem effectively. For very large sets of data, the problem takes on overwhelming proportions in terms of computer storage and the time needed to reach a satisfactory surface fit. This consideration provides the motivation behind the development of a way to pare the problem down to a more manageable size with more feasible computational times, and provides the impetus for this thesis effort.

Surface fitting of irregularly spaced data can be approached in one of two ways: by interpolation or by approximation. This theory is concerned with the determination of functions on the basis of certain functional information which is known in the form of discrete data points, (x_i,y_i,f_i) [Ref. 1:p. 7]. We can think of the dependent variable, f_i, as arising from some underlying, but not necessarily known function $f(x,y)$.

We distinguish between approximation and interpolation in surface fitting. In approximation, we wish to construct a function F which approximately fits the data; this process is generally employed when the data collection is subject to measurement error, as most data is. Interpolation, on the other hand, leads to a surface fit which matches the given data points exactly. In this case, we desire a function F

which will reproduce the original data on the constructed surface where it is assumed that the data points are very precise. [Ref. 2:pp. 203-204]

There are numerous schemes available for both interpolation and approximation as outlined by Schumaker [Ref. 2:pp. 203-268], and tested and compared by Franke [Ref. 3:pp. 181-200]. The methods can also be classified as to how they treat the given data: that is, globally or locally. Local methods are those where the value of the constructed surface F at a particular point (x,y) depends only on the data at relatively nearby points. In global methods, the value of the surface is affected by all the points. [Ref. 2:p. 204]

B. OVERVIEW

The overall objective of this thesis is to build a computer program which can be used by any researcher dealing with surface representation. The program and implementing instructions can be obtained by contacting Professor Richard Franke at the Naval Postgraduate School, Department of Mathematics, Monterey, Califonia, 93943.

This thesis incorporates a broad and diverse range of mathematical subjects, including linear algebra and matrix computations, interpolation and approximation theory, real and linear functional analysis, and numerical analysis. Most of the necessary background is provided in the first chapter, and the rest of what is needed is explained as it is required by the discussion.

Following the general background and problem description in Chapter 1, the theory of three types of splines is examined in Chapter 2. We also review the literature on the thin plate spline in terms of the

reproducing kernel function and its relation to representers of linear functionals in a Hilbert space setting in Chapter 2. Chapter 3 contains the essence of the knot selection process with a detailed exposition in one dimension. Each of the various subroutines developed for this thesis is detailed in Chapter 4. The pertinent linear algebra and matrix computation material accompanies the subroutines. Finally, in the last chapter, this surface fitting method is compared to other methods, and we see how 'representative' the knot selection process can be.

C. SPLINES, IN GENERAL

Since surface fitting is intimately involved in the problem considered here, we begin by defining some terminology. We also mention three of the prominent approaches which have been used in attempting to solve the problem of fitting curves to sets of one dimensional data, namely the interpolating spline, the smoothing spline, and the least squares spline.

In one dimension, a physical spline is a flexible strip of material which can be held fixed by weights, so that it passes through each of the given points, but goes smoothly from each interval (between the points) to the next according to the laws of beam flexure [Ref. 4:p. 199]. The mathematical approximation via cubic spline interpolation, for example, uses different cubic polynomials between successive pairs of data points. Smoothness is incorporated into the spline since both the first and second derivatives of adjacent cubics agree at each data point [Ref. 5:p. 203].

A set of linearly independent functions which span a given function space comprise a basis for that space. This means that every function in the space can be expressed in one and only one way as a linear combination of the basis functions. Furthermore, the coefficients in the expansion are uniquely determined by the basis functions, which are defined at each of the data points. [Ref. 6:p. 67]

The term 'knot' originally referred to the points at which two adjacent cubic polynomials joined in cubic spline interpolation; these points are the data points as well. We shall see that in some approximation methods, these points do not necessarily coincide and, hereafter, we shall distinguish between them. There is also an obvious relationship between these knot points and the basis functions since, as we can see with cubic spline interpolation, the basis functions depend on the locations of the knot points. Hence, we shall refer to the basis functions as being associated with the knot points, a relationship which can be understood in terms of the cubic spline interpolation example.

The interpolating spline minimizes some pre-defined functional which is a measure of the smoothness of the approximating function. The minimization is over a certain class of functions which interpolate the given data. A smoothing spline minimizes a linear combination of the same functional and a discrete term which measures the fidelity of the approximating function to the given data; here, the data is not exactly reproduced. This minimization is over a similar class of functions, but interpolation is not required. Both of these approximation methods have the same set of basis functions, because the knot points in both methods correspond to the given data points.

In least squares spline approximation, the discrete term constitutes the measurement of the error entirely. There are more data points than there are basis functions, so that the set of basis functions is derived from a smaller class of functions. The significant difference between the least squares method and the other methods lie in its use of a different, smaller set of basis functions corresponding to a set of knot points which is different from the original data.

The method chosen to fit the surface in this thesis is the global least squares Thin Plate Spline (TPS) approximation method. As a least squares spline approximation method, it involves a fewer number of basis functions than the number of data points given. The basis functions are associated with a different, smaller set of points, and the error is minimized as the discrete term alluded to above.

D. PROBLEM DESCRIPTION

Conceptually, the problem is easily understood and simply stated: Given a large set of data points (x_i, y_i, f_i), $i=1,\ldots,N$, find a significantly smaller set of knot points $(\hat{x}_j, \hat{y}_j)$, $j=1,\ldots,K$, which represents the large set reasonably well. This could be accomplished by choosing a subset of the original set, or by some process which produces a more representative set. By the phrase 'represent the large set reasonably well,' we mean that each data point is 'close' to a knot point.

Because the approximation method to be used involves the thin plate spline, each of the so-called knot points of the representative set has an associated basis function, so that the knot selection process can thought of as a way of specifying a special set of basis functions. The

ultimate goal is to approximate the surface from which the original set of data came by a thin plate spline using a significantly smaller set of points. Hence, a surface fitted to the large set and a surface fitted to the small set should essentially be the same surface.

We note that this particular approach to the problem is not totally original as referenced in the personal notes of Richard Franke at the Naval Postgraduate School. In fact, the problem was proposed by Gregory M. Nielson and Richard Franke at the Istituto per le Applicazioni della Matematica e dell' Informatica in Milan, Italy, in 1983, and some of the ideas in this effort originated there in discussions between Licia Lenarduzzi, Florencio Utreras, Nielson and Franke. However, no real progress has been reported since that time.

Throughout this thesis, a key underlying assumption is made regarding the large set of data from which the smaller set evolves. We assume that the data collector took the data so that the density of data points in a particular region is indicative of how the surface is behaving in that region. Hence, where many data points that are closely spaced occur, the surface is assumed to be active and changing, whereas sparse occurrences of the data points indicate a surface which is inactive and relatively stable. Finally, we note that a considerably more difficult problem arises when we want to find both the size and location of some smaller set of knot points; this is not the problem considered here. We are assuming that the user has decided how many knot points he considers viable for his particular application.

E. GENERAL METHODOLOGY

The net result of this thesis is a package of FORTRAN subroutines which can be collected together under a user provided 'driver' program to perform the knot selection process and generate a regular grid of points on the approximating surface. Collectively, these subroutines are referred to as the 'KSLSTPS package' (for Knot Selection for approximation using the Least Squares Thin Plate Spline). At this point, a general overview of the package is presented.

Given the number of knot points to be used, the KSLSTPS package is capable of internally generating the coordinates for a uniformly distributed set of initial guess knot points. This is accomplished by the subroutine KNTIG (for knot initial guess). Alternatively, the package user may specify his own set of initial guess knot points.

We note that solving the interpolation problem will involve a system of equations with an equivalent number of unknowns. Computationally, this may evolve into a very ambitious task, since there will be at least as many equations as there are original data points. Solving the approximation problem will also involve as many equations as there are original data points, but the number of unknowns will be significantly fewer, depending on the number of knot points used. This leads to an overdetermined system and a least squares approach in solving it.

The most important input is the comparatively large set of data points which has been collected by the package user. Typically, the number of data points is envisioned to range from 100 to several

thousand, so that it should now be apparent why an approximating surface is more feasible than one that is interpolated: the system is just too big! Additionally, since most real world data is subject to random measurement error, approximation is the preferred method in the representation.

Once the data has been provided and an initial guess for the knot points generated, the KSLSTPS package optimizes the location of the knot points so that each of the knot points represents a nearly equal number of data points subject to the constraint that the distance between these two finite point sets is minimized. These tasks are performed by the subroutines MINORM (for minimize the 2-norm) and TWEEK (for tweak the knot points around).

At this stage of the KSLSTPS package, it is necessary to solve the overdetermined system of equations that precipitates from this least squares approximation method, given the optimized knot point locations and the original data set. The subroutine LSCOEF (for least squares coefficients) performs this task using the two LINPACK subroutines, SQRDC and SQRSL, in tandem. The first, SQRDC (for Real Orthogonal Triangular Decomposition), computes the QR decomposition of the coefficient matrix, which is constructed by subroutine LSCOEF. The second, SQRSL (for Real Orthogonal Triangular Decomposition Solve) manipulates the QR decomposition to compute the least squares solutions (to be explained in detail in Chapter 4).

The KSLSTPS package then generates a rectangular grid of points in the user's region of interest at which the newly formed function F is evaluated to yield the values of the approximating surface. This is

accomplished in subroutine GEVGRD (for generate and evaluate the grid). The surface can then be plotted by DISSPLA subroutines which can be tied into a driver program. The output of the KSLSTPS subroutine package consists of the following information: the initial data points, the optimized knot points, the residuals of the least squares thin plate spline method, the maximum, mean and root-mean-squared errors of the residuals, and a grid of values on the approximating function F.

II. THEORY OF THE THIN PLATE SPLINE

A. INTRODUCTION

Continuing the background development in one dimension, we examine the three types of spline methods which have a direct extension to two dimensions. Then we delve into their natural generalizations to two dimensions to complete the background picture. Finally, we present two summarized versions of how the thin plate spline came into being, and explain why it has particular application to our problem.

B. ONE DIMENSIONAL SPLINES

The classical example of an interpolating spline in one dimension has already been mentioned in terms of the cubic interpolating spline. See Ref. 5:pp. 204-209 for a well presented description of the details of cubic spline interpolation. In one dimension, the interpolating spline minimizes a functional of the form

$$\int_{x_1}^{x_N} |F''(x)|^2 \, dx, \tag{1}$$

where F is the approximating function [Ref. 7:p. 76]. This pseudo-norm is minimized over the class of functions which interpolate the data, and which have square integrable second derivatives and absolutely continuous first derivatives [Ref. 7:p. 75]. Thus, a measure of the smoothness of the approximating function is used in this technique where we have said that the number of basis functions corresponds to the number of data points, which are the knot points.

The objective of the smoothing spline technique, where the knot points and the data points coincide in number and location, is to produce a surface whose values at the data points are close to the dependent variables, and whose errors are 'smoothed out'. As de Boor describes this problem [Ref. 8:pp. 235-239], we are given the approximate values $y_i = g(x_i) + e_i$ of some supposedly smooth function g at data points $x_1, x_2, \ldots, x_N$, and an estimate of the standard deviations, δy_i, of the errors in y_i. The object is to recover the unknown function g from these data by constructing a function $F = F_p$, such that the

$$\text{Min } p\sum_{i=1}^{N}\left[\frac{y_i - F(x_i)}{\delta y_i}\right]^2 + (1-p)\int_{x_1}^{x_N}\left[F''(t)\right]^2 dt \qquad (2)$$

is achieved, where $p \in [0,1]$ is a parameter. Here, we are dealing with a larger class of functions, which are those which have square integrable second derivatives and absolutely continuous first derivatives, but without the interpolating constraint as seen in the interpolating spline. This minimization reflects a compromise between approximating the data points as closely as possible, while maintaining a certain degree of smoothness in the function. The choice of parameter p dictates which of these characteristics is afforded more consideration.

There are at least two approaches to take in solving the least squares spline problem, wherein we have a larger set of data points and a smaller second set of distinct knot points. Here, we wish to choose a function F so as to minimize the expression

$$\sum_{i=1}^{N}\left[F(x_i) - f_i\right]^2/\sigma_i^2, \qquad (3)$$

where σ_i^2 is the variance at the i^{th} data point. One option is to fix the knot points according to some prearranged criterion in advance of the minimization. In doing so, we may be applying the assumption stated earlier, wherein the locations of the data points reflect the behavior of the underlying function (i.e., high density implies an active surface; low density implies an inactive surface), or some other criterion may be used.

The other option is to choose the knot point locations in connection with the actual behavior of the dependent variable value, f_i. This leads to serious consequences in terms of the non-linearity of the normal equations which result in such a minimization. We note, for example, that the use of cubic spline interpolation basis functions to solve the least squares approximation problem by allowing the knot point coordinates to be an added parameter in each of the equations has been attempted in one dimension. This minimization involves non-linear terms in each of the knot points since the basis functions depend on the knot points. Problems also arise in terms of non-uniqueness of solutions and in the coalescing of knot points. [Ref. 8:p. 271]

C. TWO DIMENSIONAL SPLINES

The interpolating thin plate spline in two dimensions involves the minimization of the functional

$$\iint_{R^2} \left[\left(\frac{\partial^2 F}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 F}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 F}{\partial y^2} \right)^2 \right] dxdy, \qquad (4)$$

where R^2 is the real plane [Ref. 2:p. 215]. The minimization is performed over the space of all Schwarz distributions with square integrable second derivatives [Ref. 2:p. 215], so that the function

space involved is infinite dimensional, a distinct advantage. As in one dimension, the basis functions are associated with each of the data points, so that a large system of equations must be solved when the number of data points is large. Hence, the method is not used for systems involving more than about 200 data points.

The two dimensional analog of the smoothing spline problem becomes that of finding the minimum of

$$\frac{1}{N}\sum_{i=1}^{N}\left[F(x_i,y_i) - f_i\right]^2/\sigma_i^2 + \lambda\iint_{R^2}\left[\left(\frac{\partial^2 F}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 F}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 F}{\partial y^2}\right)^2\right]dA, \quad (5)$$

where σ_i^2 is the variance of the error at the i^{th} point, F is the approximating spline, and λ is the smoothing parameter. This equation is a linear combination of two terms, where the first indicates how closely the data is approximated, while the second controls the degree of smoothness. We note that when the smoothing parameter approaches zero, the smoothing spline function becomes an interpolating spline. Furthermore, when the parameter grows large, the second derivatives must vanish and the problem reduces to a least squares approximation by a linear function.

Wahba and Wendelberger describe a method known as generalized cross-validation (GCV) to determine the value of the smoothing parameter [Ref. 9:pp. 1122-1143]. The basic idea of GCV can be understood by first describing simple cross-validation. By fixing all but the i^{th} data point, we construct an approximating surface which is used to find a predicted value, p_i, for the dependent variable, f_i, at that point. The parameter is set arbitrarily, and this 'fixing' is done for each

data point separately. There, the value of the simple cross-validation (CV) function is

$$CV(\lambda) = \sum_{i=1}^{N} \left(f_i - p_i \right)^2$$

and λ is chosen to minimize this sum. The evaluation of the sum is an expensive calculation, and the GCV method is a simpler way of approximating the minimizing value of the parameter.

While the idea behind the smoothing spline method with GCV is appealing, the application of it to large sets of data is not feasible. This results from the fact that each of the data points has a basis function associated with it leading to the solution of a N+3 by N+3 system of linear equations. Manipulation of such systems quickly exceeds the capability of most computers when more than 200 data points are involved.

The third two dimensional scheme, the least squares thin plate spline method, involves the minimization of the discrete term which comprises the totality of the error. The error is

$$\frac{1}{N} \sum_{i=1}^{N} \left[F(x_i, y_i) - f_i \right]^2 / \sigma_i^2,$$

where F is a function of two independent variables (x,y). The function space of F is a finite dimensional subspace of the function space pertaining to interpolating and smoothing thin plate splines. As we shall see, the basis functions used are linear combinations of functions no more complicated than $d_i^2 \cdot \log(d_i)$ plus some linear terms, where

$$d_i^2 = (x - x_i)^2 + (y - y_i)^2.$$

We note that, as expected, each of the basis functions is associated with one of the knot points, and that we will be using fewer basis functions than the number of given data points.

Given the analogies between this method in one and two dimensional space, it seems reasonable to attempt to solve the least squares approximation problem using the basis functions of the thin plate spline associated with the knot point coordinates as additional parameters in the minimization process. This approach leads to serious nonlinear complications, and has not been attempted here. Instead, the approach taken in this thesis involves performing the knot selection process separately, in advance of the minimization.

D. THE THIN PLATE SPLINE, IN DEPTH

Interpolation of scattered data by the thin plate spline (TPS), or surface spline, under point loads was originated by Harder and Desmarais using engineering considerations [Ref. 10:pp. 189-191]. Suppose we have an infinitesimally thin plate of infinite extent that can be deformed in bending only. The differential equation

$$D\left[\left(\frac{\partial^4 W}{\partial x^4}\right) + 2\left(\frac{\partial^4 W}{\partial x^2 \partial y^2}\right) + \left(\frac{\partial^4 W}{\partial y^4}\right)\right] = q, \qquad (7)$$

where D depends on the properties of the plate material, W is the lateral deflection, and q is the lateral load, relates the bending deflections to point loads on the plate. The problem becomes one of finding the point loads which, when applied to the plate, cause deflections in the plate and force the plate to pass exactly through the data points.

The basis functions for the solutions to Equation (7) are found by introducing polar coordinates and integrating Equation (7) for one point load centered at the origin. This yields the fundamental solution

$$W(d) = C + Bd^2 + (P/16\pi D)d^2 \cdot \log(d), \qquad (8)$$

where C and B are the constants of integration, and P is the point load.

The TPS function is obtained by superposition of the deflections due to all of the point loads, giving

$$W(x,y) = \sum_{i=1}^{N} [C_i + B_i d_i^2 + (P_i/16\pi D)d_i^2 \cdot \log(d_i)], \qquad (9)$$

where

$$d_i^2 = (x - x_i)^2 + (y - y_i)^2.$$

By considering what happens to the spline at long distances from the point loads (large d values), and applying the usual engineering constraints of a rigid body in equilibrium (i.e., no translation and no rotation about either axis), the equation can be rearranged into a form that is useful for computation,

$$W(x,y) = a_o + a_1 x + a_2 y + \sum_{i=1}^{N} A_i d_i^2 \cdot \log(d_i), \qquad (10)$$

where

$$A_i = P_i/16\pi D,$$

$$a_o = \sum_{i=1}^{N} [C_i + B_i(x_i^2 + y^2)],$$

$$a_1 = -2 \sum_{i=1}^{N} B_i x_i \qquad \text{and} \qquad a_2 = -2 \sum_{i=1}^{N} B_i y_i. \qquad (11)$$

An alternative approach is to consider the geometric effect of the equilibrium conditions, which is to eliminate all but the linear terms

[Ref. 3:p. 191]. The simplification in Equation (10) relies in part on the regularity condition which is analogous to boundary conditions on a finite domain. The equilibrium conditions are:

$$\sum_{i=1}^{N} P_i = 0 \qquad \text{(no translation)}, \tag{12}$$

$$\sum_{i=1}^{N} x_i P_i = 0 \qquad \text{(no moments in the x-direction)}, \tag{13}$$

and

$$\sum_{i=1}^{N} y_i P_i = 0 \qquad \text{(no moments in the y-direction)}. \tag{14}$$

Thus, we see the deflection can be described as a set of linear combinations of the basis functions 1, x, y, and $d_i^2 \cdot \log(d_i)$, where the coefficients a_0, a_1, a_2, and A_i must be found through solution of the system of equations. We note that there are a total of N+3 equations in N+3 unknowns, as would be expected in an interpolation problem. The number of equations correspond to one for the deflection at each data point for a total of N, and the three equilibrium Equations (12) - (14).

The unknowns are the coefficients a_0, a_1, a_2, and A_i, i=1,...,N. The deflection values are the dependent variables, labeled f_i, at each of the data points (x_i, y_i). In matrix notation, the system of equations may be written as

$$\begin{bmatrix} 0 & d_{12}^2 \log d_{12} & \cdots & d_{1N}^2 \log d_{1N} & x_1 & y_2 & 1 \\ d^2_{21} \log d^2_{21} & 0 & \cdots & d^2_{2N} \log d^2_{2N} & x_2 & y_2 & 1 \\ \vdots & & & & \vdots & \vdots & \vdots \\ d^2_{N1} \log d_{N1} & d^2_{N2} \log d_{N1} & \cdots & 0 & x_N & y_N & 1 \\ x_1 & x_2 & \cdots & x_N & 0 & 0 & 0 \\ y_1 & y_2 & \cdots & y_N & 0 & 0 & 0 \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \\ a_1 \\ a_2 \\ a_0 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where

$$d_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2 \qquad i, j=1,\ldots,N.$$

Another attractive aspect behind the use of TPS is the existence of an elegant mathematical theory developed by Duchon [Ref. 11] and Meinguet [Ref. 12:pp. 127-142]. First, we digress to summarize some of the work done in the context of a Hilbert Space as it applies to our particular application and to establish the relationship between the reproducing kernel function and the representers of linear functionals.

A Hilbert space is an infinite dimensional inner product space which is complete in the norm generated by the inner product [Ref. 13:p. 92]. It is an abstract concept which can be made concrete using examples such as those cited in Ref. 1:pp. 203-214. We are interested in the Hilbert space consisting of all tempered distributions f on R^2 whose Fourier transforms $\hat{f}$ satisfy

$$\iint_{R^2} |\hat{f}|^2 \, dt < \infty,$$

[Ref. 2:p. 216]. A full discussion of the reproducing kernel function, including its definition and application theorems for various example Hilbert spaces, can be found in Ref. 1:pp. 316-326.

The significance of this discussion revolves around what can be said about representers and the optimal approximation of linear functionals in a Hilbert space. We are given the values of the bounded linear functionals, $L_i(f) = f(Q_i)$, where Q_i denotes the data point, (x_i, y_i), and f is the underlying unknown function we wish to approximate by the function F. The optimal approximation to f is that linear combination of the representers of the $L_i(f)$, which minimizes the error as defined in Equation (4) for the particular Hilbert space. Each of the functionals has its own unique representer, and the optimal approximation can be calculated if the representers of the appropriate functionals are known. Finally, the representers can be easily determined if the reproducing kernel function for the Hilbert space is known. A particularly good discussion of the details of the derivation of the unique representer for our particular functional is presented in Ref. 9:pp. 1138-1140. Then the coefficients in the approximation can be determined from the normal equations, which are equivalent to the interpolation conditions. [Ref. 14:pp. 115-116]

The theory of Duchon and Meinguet, was summarized by L. L. Schumaker [Ref. 2:pp. 214-216], as follows. Let Θ be a functional on a linear space X which measures the smoothness of an element g in X; call U the set of smooth functions in X which interpolate g. Thus,

$$U = \{g \in X : g(x_i, y_i) = f_i,\ i = 1, \ldots, N\}.$$

We specify the convention that the smaller Θ is, the smoother g is, so that our problem becomes one of finding the function $g \in U$ which minimizes $\Theta(g)$.

We further specify that X be a certain semi-Hilbert space, where the semi-norm on X is defined by

$$||g||^2 = \Theta(g).$$

Let the class of functions η, whose semi-norm vanishes, be

$$\eta = \{g \in X : ||g|| = 0\}$$

Duchon has shown that (under some additional conditions on X), the minimization problem always has a solution that is unique up to an element of η (i.e., they differ by only a linear term for the linear functional we will want to consider). We note that the space is semi-Hilbert in the sense that the semi-norm fails with respect to the definiteness property of normed linear spaces.

Furthermore, there exists a reproducing kernel K, such that

$$F(x,y) = \sum_{i=1}^{N} a_i K[(x,y);(x_i,y_i)] + \sum_{i=1}^{m} b_i p_i(x,y), \qquad (15)$$

where the functions, $\{p_i\}_1^m$, form the basis for η. We note that the $K[(x,y);(x_i,y_i)]$ terms are the representers of the functionals evaluated at the points (x_i,y_i). Also, the semi-norm of g is unaffected by the inclusion of the linear combination of basis functions, $\{p_i\}_1^m$, since by definition $||p_i|| = 0$. As before, the coefficients $\{a_i\}$ and $\{b_i\}$ can be determined through solution of the system of linear equations,

$$\sum_{j=1}^{K} a_i K[(x_j,y_j);(x_i,y_i)] + \sum_{i=1}^{m} b_i p_i(x_j,y_j) = F_j, \quad j=1,\ldots,N, \qquad (16)$$

with the additional orthogonality conditions

$$\sum_{i=1}^{N} a_i p_k(x_i,y_i) = 0, \quad k=1,\ldots,m.$$

At this point, the key question is resolution of the appropriate reproducing kernel function. Consider

$$\Theta(f) = \iint_{R^2} \left[\left(\frac{\partial^2 F}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 F}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 F}{\partial y^2}\right)^2 \right] dxdy,$$

where X is the space of all functions on the infinite domain, H, which have Schwarz distributions whose second derivatives are square integrable. Duchon has shown that it is possible to obtain explicit expressions for the reproducing kernel, and these have the form of the sum of $d_i^2 \cdot \log(d_i)$ terms and some additional linear terms.

In this case, the interpolating spline solution of the minimization problem takes the form

$$F(x,y) = \sum_{i=1}^{N} a_i d_i^2(x,y) \log[d_i(x,y)] + b_1 x + b_2 y + b_3, \qquad (17)$$

where

$$d_i^2(x,y) = (x-x_i)^2 + (y-y_i)^2.$$

As was mentioned earlier, the coefficients are determined from the system of linear equations with $m = 3$, n = span $(1,x,y)$, and $K(P,Q)$, a linear combination of $d_i^2 \cdot \log(d_i)$ terms and functions of n. We note the similarity between Equation (10), derived by the mathematicians, and Equation (17), developed by the engineers; that is, they are the same.

E. PERSPECTIVE

Now, we make an argument for using the same basis functions found in the TPS interpolation method as a logical extension for their use in TPS approximation. In one dimension (one independent variable), we mentioned the established fact that the basis functions in cubic spline interpolation can also be used in cubic spline approximation. Hence, it is not unreasonable to attempt to extend this idea into two dimensions; namely, we can expect the basis functions found in TPS interpolation to be valid in TPS approximation. Herein lies the rationale for the use of least squares TPS in the surface construction process.

III. THE KNOT SELECTION PROCESS

A. INTRODUCTION

Most of the research for this thesis involved the creation and subsequent testing and modification of two FORTRAN subroutines which could be used to answer the following questions:

1) By what criteria should the distance between two finite point sets be measured?

2) What means should be employed to generate the locations of the knot points to effectively represent the original data set? and

3) How should the knot points be moved around the region of interest to insure that a reasonable configuration of knots can be found?

We wished to base the answers to these questions on experimental evidence. In the process of the investigation, other questions arose including:

1) Is there a unique set of knot points which minimizes the distance between two finite point sets?

2) If so, how can it be found? and

3) How does one know it has been found?

Although there are still some remaining open questions, this chapter describes what we do know about the problem in light of the answers to some of the questions posed above. Specifically, the simple mechanism by which the knots are located is explained and its derivation is given. Then, we give a theorem concerning how a certain measure of the distance between two finite point sets behaves when the representation is accomplished in a certain way. In the interest of understanding what is happening in this multi-dimensional problem, we investigate its one

dimensional version. Finally, we discuss a modification to the knot selection process which provides for a more equitable representation of the data points by the knot points.

B. LOCATING A KNOT

As the discussion progresses, it will be useful to have some basic definitions in mind, as well as some ideas about how one might consider measuring the distance between two finite point sets. We wish to represent a large set of data (x_i, y_i), $i=1,\ldots,N$, using a significantly smaller set of points, $(\hat{x}_j, \hat{y}_j)$, $j=1,\ldots,K$. Suppose we have two finite point sets:

$$P = \{(x_i, y_i),\ i=1,\ldots,N\}$$

and

$$Q = \{(\hat{x}_j, \hat{y}_j),\ j=1,\ldots,K\},$$

where we assume $K < N$. Define a vector V of dimension N, whose elements are the Euclidean distances measured from each of the data points to the closest neighboring knot point. That is

$$v_i = \sqrt{\min_{1\le j\le K}[(x_i - \hat{x}_j)^2 + (y_i - \hat{y}_j)^2]}, \quad i=1,\ldots,N. \tag{1}$$

It is important to note that for every point in the larger set P, we find the closest point in the smaller set Q, so that we are minimizing the sum of the minimum individual distances (between the points in the larger set and the points in the smaller set), over the number of points in the smaller set. This leads to a process of $N \cdot K$ measurements, and results in a vector with dimension $N = \max(N,K)$.

Next, define a norm [Ref. 1:p. 129], in the usual way, to be the real number, denoted $||\ \ ||$, assigned to each vector V in the vector space such that

$$||V|| = \sum_{i=1}^{N} |v_i|^{1/p}, \quad p=1,2,... \qquad (2)$$

and satisfying the properties of a normed linear space, which are:

positivity, $||V|| > 0$;

definiteness, $||V|| = 0$ iff $V=0$;

homogeneity, $||\alpha V|| = |\alpha| \cdot ||V||$; and the

triangle inequality, $||V|| \leq ||U|| + ||T||$,

where U, T and V are vectors, and α is a scalar. Furthermore, we can define the distance between two finite point sets, d(P,Q), as the norm of the vector V, which is formed as described in Equation (2) above. Thus,

$$d(P,Q) = ||V||.$$

The criterion by which the 'distance' between two finite point sets is measured, is the value of the 'global norm' (GN), which is defined to be the square root of the sum of the squares of the Euclidean distances between the data points 'belonging' to a particular knot point and that knot point. This GN corresponds to Equation (2) for p = 2; as the 2-norm between two finite point sets, its square will be easier to work with.

A tile is defined to be that region of the plane containing the points in the plane which are closer to one particular knot point than to any other, subject to a tie breaking criterion described in the next chapter. A tile is found by drawing the perpendicular bisectors to the lines joining each of the knots, extending the bisectors until they

intersect. Each of the tiles is thus a convex region of the plane, and when this partitioning of the plane is considered as a whole, it is termed a Dirichlet Tesselation. Figure 3.1 illustrates a Dirichlet Tesselation, where we note that each tile is associated with one knot point which, in turn, 'owns' one or more data points.

The knot locating process can best be pictured as occurring in two distinct steps, each involving a separate numerical quantity. The first quantity, the GN^2 value, has already been introduced; the second, the global tile difference (DSUM), is defined to be the sum of the squares of the differences between the number of data points in each tile and the average number of points per tile, the ratio of the number of data points, N, to the number of knots (tiles), K. In the last section, we will see how the DSUM value comes into play in the knot locating procedure.

The least squares criterion is used for determining the location of the knot point within its tile. The least squares derivation is as follows: Minimize

$$S_j = \sum_{i \in I_j} [(x_i - \hat{x}_j)^2 + (y_i - \hat{y}_j)^2],$$

where $I_j = \{i:(x_i,y_i)$ is closer to $(\hat{x}_j,\hat{y}_j)$ than any other $(\hat{x}_k,\hat{y}_k)\}$. There will be a total of K different functions S_j, one for each sum of the squares of the Euclidean distances between the data points corresponding to indices in I_j and a particular knot to which they are closest. For each component $\hat{x}_j$ and $\hat{y}_j$, a necessary condition for optimization is that each of the partial derivatives $\partial S_j/\partial \hat{x}_j$ and $\partial S_j/\partial \hat{y}_j$ equal zero, yielding the equations

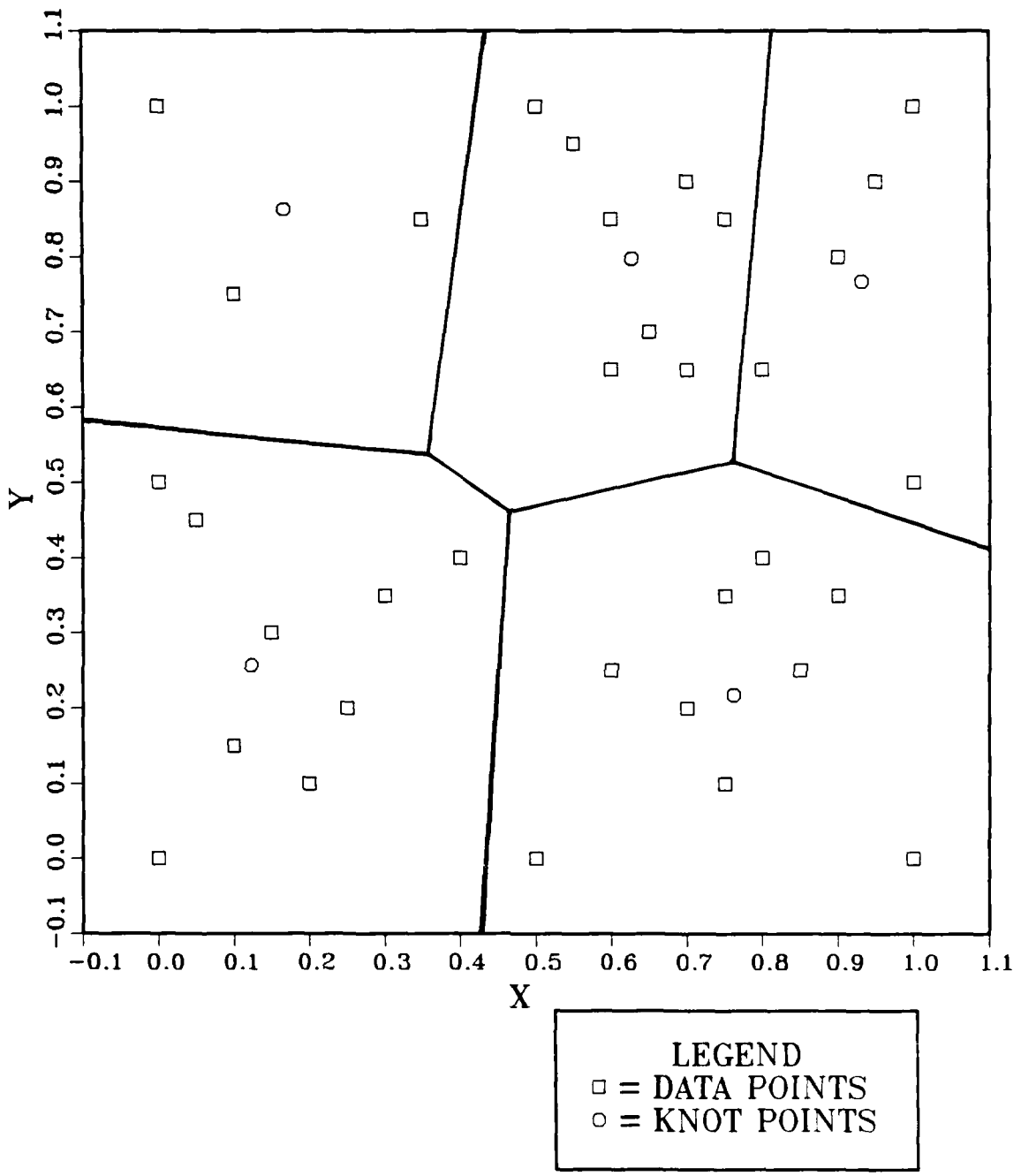


Figure 3.1. A Dirichlet Tesselation with 5 Tiles.

$$\frac{\partial S_j}{\partial \hat{x}_j} = -2 \sum_{i \epsilon I_j} (x_i - \hat{x}_j) = 0$$

and

$$\frac{\partial S_j}{\partial \hat{y}_j} = -2 \sum_{i \epsilon I_j} (y_i - \hat{y}_j) = 0.$$

These equations can be written as

$$-2(\sum_{i \epsilon I_j} x_i - m_j \hat{x}_j) = 0 \qquad \text{and} \qquad -2(\sum_{i \epsilon I_j} y_i - m_j \hat{y}_j) = 0,$$

or

$$\hat{x}_j = \sum_{i \epsilon I_j} x_i / m_j \qquad \text{and} \qquad \hat{y}_j = \sum_{i \epsilon I_j} y_i / m_j,$$

where m_j is the number of indices in each set I_j. We conclude that this process yields the knot point coordinates which will minimize the contribution from a particular tile, and that these knot point coordinates correspond to what we have termed the centroid (with respect to the data points in the tile) of each tile in the Dirichlet Tesselation.

C. THEOREM ONE

We have stated that the knot selection process can best be understood in the context of two distinct stages, each involving a separate quantity. First, we shall consider the process described in Section B, concerning how a knot is moved so that it occupies the centroid of its tile. Then, we examine the effects on the Dirichlet Tesselation of having moved the knot. Suppose we have a large set of fixed data points and some 'uniformly' distributed initial set of knot points with which we represent the data points. We specify that we will measure the distance between these two finite point sets using the GN^2 we defined

earlier. Consider the Dirichlet Tesselation associated with this particular configuration of knot points and suppose we further specify that we will move the knot points so that each one occupies the center position with respect to the data points in its tile, the centroid.

THEOREM ONE. The value of GN² is a monotonically decreasing function of each iteration involving a knot movement. In other words, each time the moving process occurs in accordance with the centroid specification (constraint), the value of GN^2 will either decrease or will remain the same.

PROOF: Assign the data point indices to the K sets I_j, $j=1,\ldots,K$, according to which of the K knots a particular data point is closest. Thus,

$$I_j = \{i:(x_i,y_i) \text{ is closer to } (\hat{x}_j,\hat{y}_j) \text{ than any other } (\hat{x}_k,\hat{y}_k)\}.$$

We wish to find the set of knot points which minimize the GN^2 value which is

$$GN^2 = \sum_{j=1}^{K} \sum_{i \in I_j} [(x_i - \hat{x}_j)^2 + (y_i - \hat{y}_j)^2]. \qquad (3)$$

This minimization amounts to a least squares optimization procedure which was detailed in the last section. The easiest way to describe the minimization is to think of it as occuring in two separate steps.

First, consider what happens to the individual terms of the interior sum when the centroid constraint is applied. By moving the knot point, $(\hat{x}_j,\hat{y}_j)$, of a particular tile to the centroid of the tile (with respect to the data points, not the tile area), each of the interior sums is minimized, and hence, the exterior sum is also minimized. As this is

accomplished, we have a new set of knot points, designated $(\hat{x}'_j, \hat{y}'_j)$, and so we may write the intermediate value of the sum as

$$GN^2_{IN} = \sum_{j=1}^{K} \sum_{i \epsilon I_j} [(x_i - \hat{x}'_j)^2 + (y_i - \hat{y}'_j)^2]. \qquad (4)$$

Obviously, the intermediate value of GN^2 is less than or equal to the previous value of GN^2; they are equal when the original knot points occupy the centroids of the tiles to begin with.

Secondly, now that one or more of the knot points have been moved to occupy the centroids of their tiles, the boundaries of the Dirichlet Tesselation have also changed. So we must consider what happens when one knot point is moved, so that one or more of the data points 'belonging' to it now 'belongs' to a different knot point. This means that the compositions of the sets I_j have changed so that new sets I'_j are now incorporated into the minimization. The new GN^2 value, designated GN'^2, is

$$GN'^2 = \sum_{j=1}^{K} \sum_{i \epsilon I'_j} [(x_i - \hat{x}'_j)^2 + (y_i - \hat{y}'_j)^2]. \qquad (5)$$

As part of the changes in the compositions of the sets I_j, we know that some tiles gained one or more points, and that some tiles lost one or more points.

If we look at Equations (4) and (5) as sums of N terms, we see that there is one term for each data point. When a particular data point is associated with a knot point in Equation (4) different from the knot point it is associated with in Equation (5), it is because the data point is closer to the new knot point than it is to the old knot point. Consequently, the term for that data point is smaller. We know this

because the point is now closer to the knot of the gaining tile than it is to the knot of the losing tile, and hence, its contribution to the overall sum is smaller.

Q.E.D.

D. A ONE DIMENSIONAL VERSION

Having proved Theorem One, we now examine a one dimensional example in order to gain some insight into how the 2K dimensional entity behaves. Since we cannot study the behavior of a function of 2K variables very easily, we wish to draw some analogies for the 2K dimensional case using this example. We can also observe the theorem given in the last section, in one dimension.

The function being optimized has as its domain, the set of ordered pairs corresponding to all possible data points, and its range can be written as

$$\sum_{i=1}^{N} \operatorname{Min}_{j}[(x_i - \hat{x}_j)^2 + (y_i - \hat{y}_j)^2].$$

We could think of it as a linear combination of K functions $g_j(\hat{x}_j,\hat{y}_j)$, j=1,...,K, selected from a larger set of N · K functions $g_i(\hat{x}_j, \hat{y}_j)$, i=1,...,N, j=1,...,K, such that $g_j = \min g_i(\hat{x}_j,\hat{y}_j)$. In this context, we see that it is a function of 2K independent variables, which are the knot point coordinates $(\hat{x}_1,\hat{y}_1,\ldots,\hat{x}_K,\hat{y}_K)$; each (x_i,y_i), i=1,...,N, is fixed as a data point.

Clearly, this function is piecewise quadratic consisting of a sum of the K minimum component quadratic functions $g_j(\hat{x}_j,\hat{y}_j)$. Since each of these is continuous, and since the minimization process does not affect their continuity, the composite function is also a continuous function

as the sum of the continuous component functions. However, we can show that the composite function is not differentiable at each point for which it is defined and hence, we can conclude that corners will occur throughout its range.

Our objective in this exercise is two-fold: first, we wish to see that the one dimensional GN^2 function is piecewise quadratic; second, we wish to see how a knot point will move so that it occupies the local minimum of the piece of the GN^2 function it happens to fall on. Figures 3.2 through 3.5 present a series of cross-sectional views of the GN^2 function under the constraint that one knot point location is fixed.

We note the piecewise quadratic nature of each of these graphs; we also observe that several local minima and maxima occur at certain locations along the abscissa. Suppose we want to represent five data points at -1, -0.5, 0, 0.33, and 2 with two knot points; in each of the graphs, we fix one of the knot points at some 'arbitrary' location. The GN^2 function we wish to optimize is composed of the five quadratics,

$$GN^2 = \text{Min}[(\hat{x}_1 + 1)^2,(\hat{x}_2 + 1)^2] + \text{Min}[(\hat{x}_1 + 1/2)^2,(\hat{x}_2 + 1/2)^2] +$$
$$\text{Min}(\hat{x}_1^2,\hat{x}_2^2) + \text{Min}[(\hat{x}_1 - 1)^2,(\hat{x}_2 - 1)^2] + \text{Min}[(\hat{x}_1 - 2)^2,(\hat{x}_2 - 2)^2],$$

where $\hat{x}_1$ and $\hat{x}_2$ are the as yet unspecified locations of the knots, or the variables on which the optimization will occur.

Analyzing the first graph (Figure 3.2) in some depth, we see the fixed knot point $\hat{x}_2 = 0.5$, and thus write the GN^2 function as

$$GN^2 = \text{Min}[(\hat{x}_1 + 1)^2, 1/4] + \text{Min}[(\hat{x}_1 + 1/2)^2,0] +$$
$$\text{Min}(\hat{x}_1^2, 1/4) + \text{Min}[(\hat{x}_1 - 1/3)^2,\frac{25}{36}] + \text{Min}[(\hat{x}_1 - 2)^2,\frac{25}{4}].$$

This function is graphed in Figure 3.2, and may also be expressed as

$$GN^2 = \begin{cases} 5.788, & \hat{x}_1 < -5/2, \\ (\hat{x}_1 + 1)^2 + 3.528, & -5/2 \leq \hat{x}_1 \leq -3/2, \\ 2(\hat{x}_1 + 3/4)^2 + 2.653, & -3/2 \leq \hat{x}_1 \leq -1/2, \\ 3(\hat{x}_1 + 1/2)^2 + 2.778, & -1/2 \leq \hat{x}_1 \leq 1/3, \\ 4(\hat{x}_1 + 7/24)^2 + 3.528, & 1/3 \leq \hat{x}_1 \leq 1/2, \\ (\hat{x}_1 - 2)^2 + 3.528, & 1/2 \leq \hat{x}_1 \leq 7/2, \\ 5.778, & \hat{x}_1 > 7/2. \end{cases}$$

The GN^2 function is a constant for any knot point locations of $\hat{x}_1$, which are further from the data points than the fixed knot point $\hat{x}_2 = 0.5$. The graph tells us the value of the GN^2 function (given one knot is fixed at $\hat{x}_2 = 0.5$), when it is evaluated at any location for the other knot, $\hat{x}_1$. For example, at $\hat{x}_1 = 0$, the GN^2 function value is 3.5. We note that when both knots are located at the same point ($\hat{x}_1 = \hat{x}_2 = 0.5$ in this case), a local maximum GN^2 value occurs; having both knots occupy the same location is a real possibility in the 2K dimensional case. However, when this occurs, it is only a temporary situation, since only one of these knots will be 'credited' with any data points in the next iteration of the MINORM subroutine. This one sided assignment process forces the other located knot to be moved to the nearest data point, as described in Chapter 4, Section B, thereby relieving the coalescing of knot points. We also observe that the points at which the

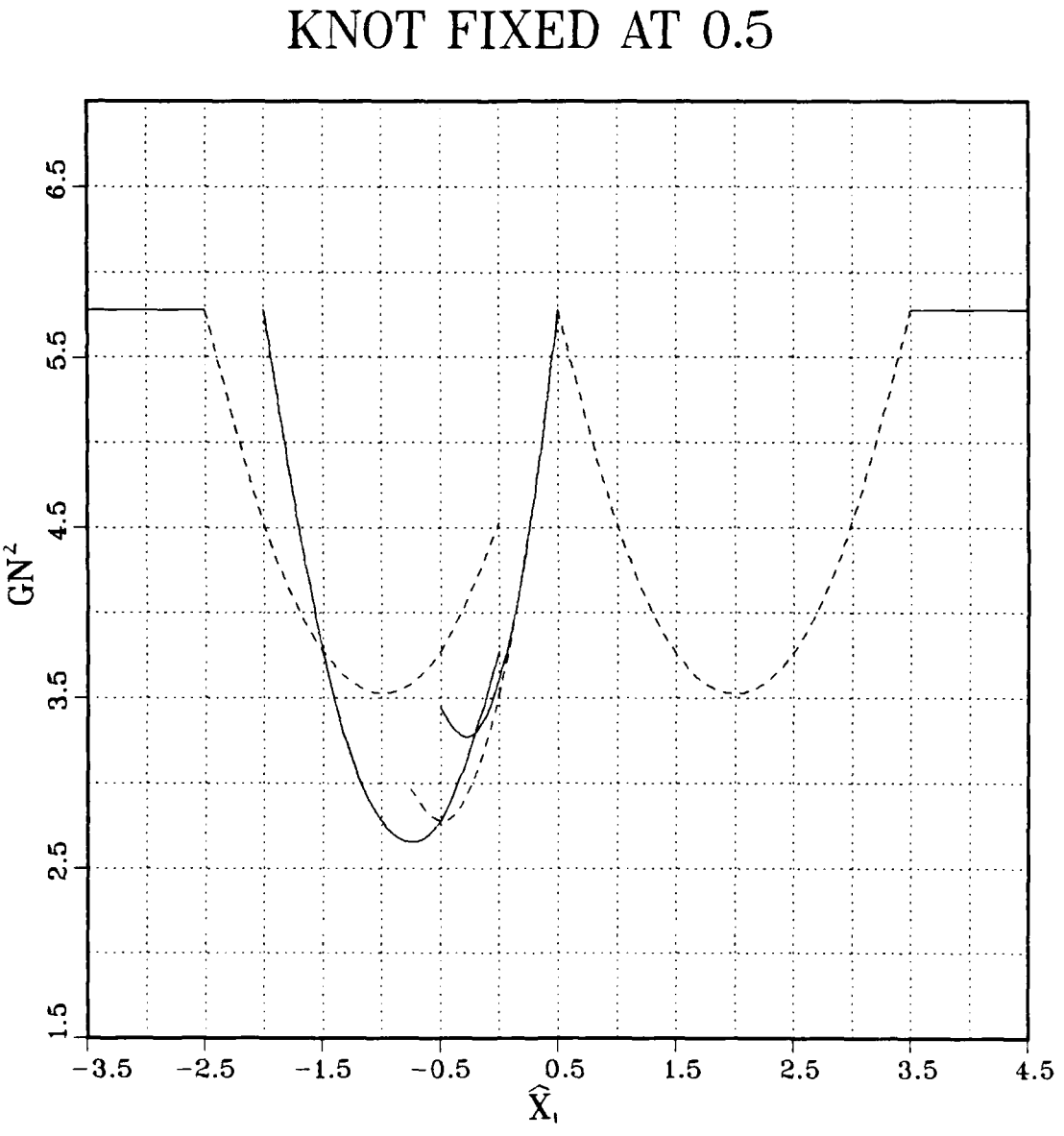


Figure 3.2. One Dimensional Cross Section A.

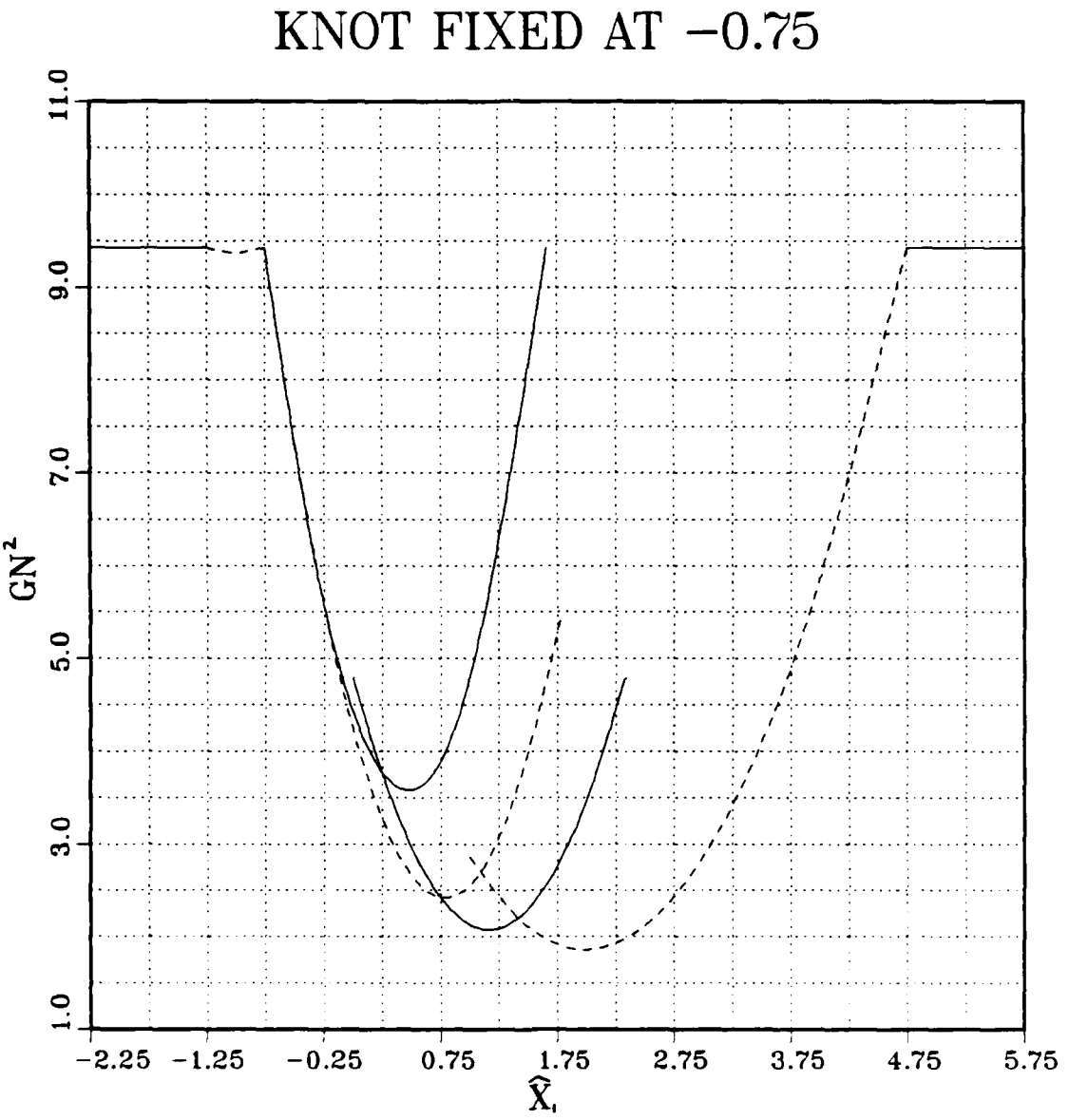


Figure 3.3. One Dimensional Cross Section B.

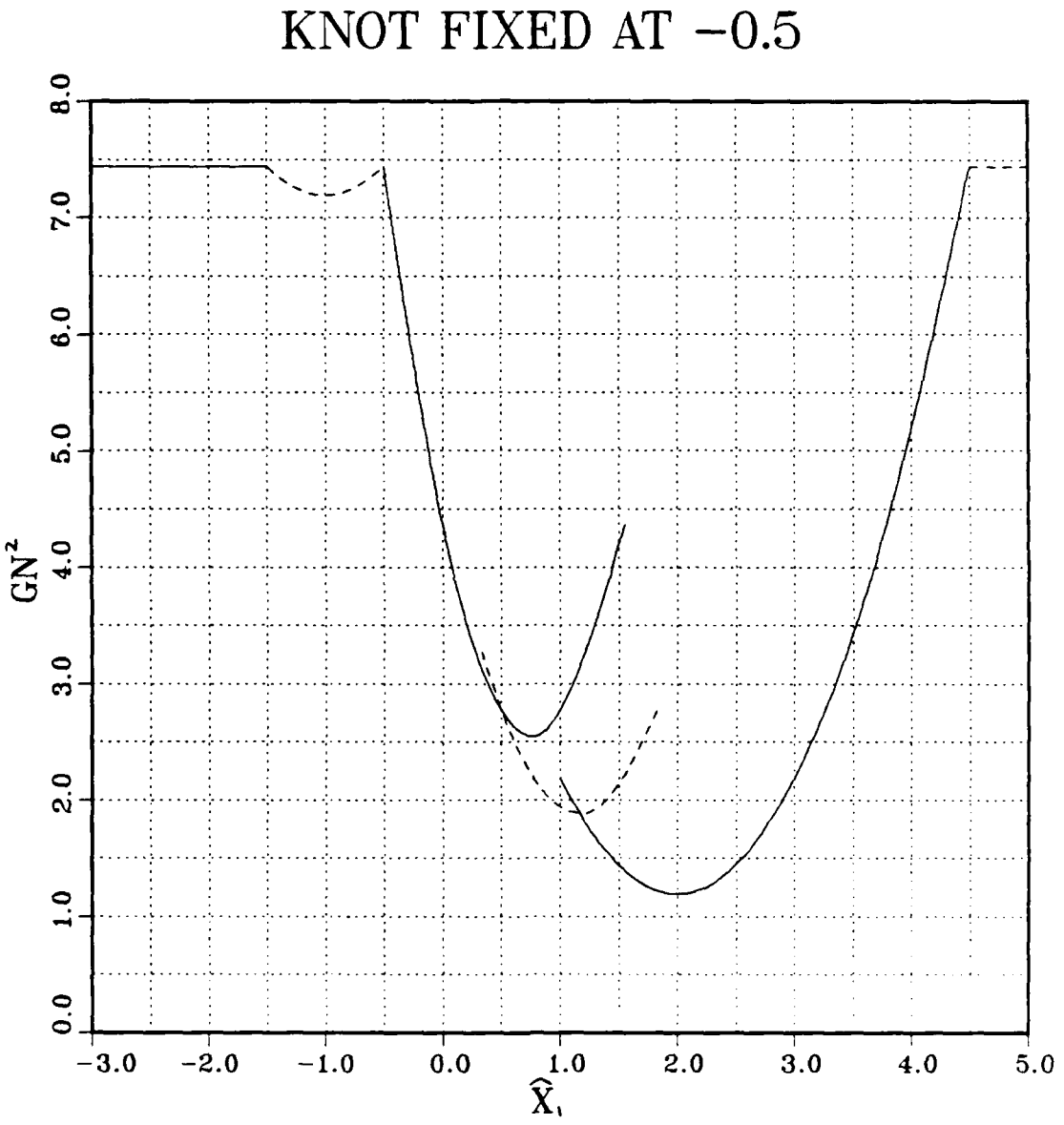


Figure 3.4. One Dimensional Cross Section C.

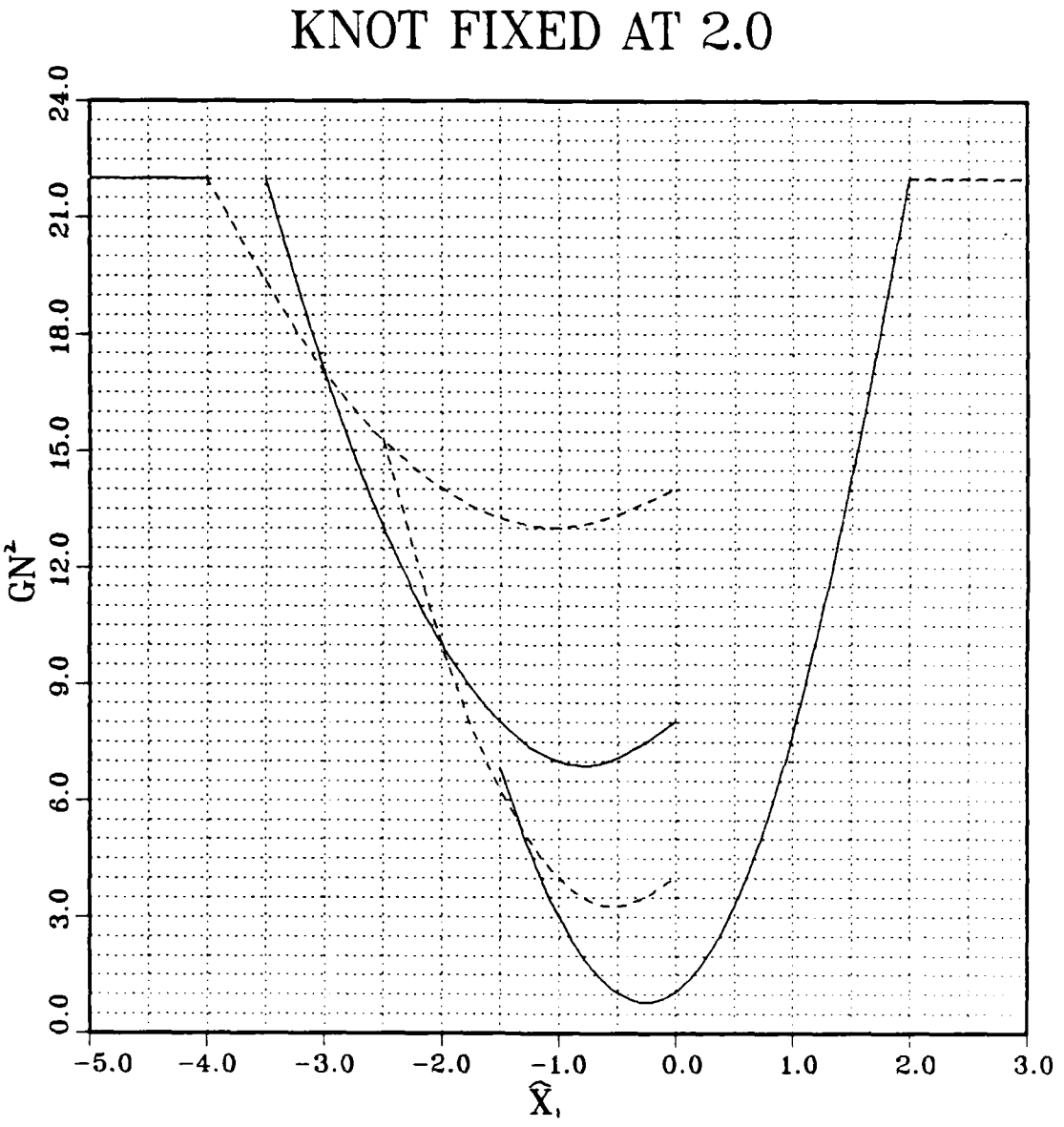


Figure 3.5. One Dimensional Cross Section D.

GN^2 function value is defined by a different quadratic expression corresponds to those points at which the tile boundary moves over a data point. In Figure 3.2, these points are -2.5, -1.5, -0.75, -0.5, 0.167, 0.5, and 3.5.

Each of these graphs may be interpreted in the following manner: for one knot fixed as indicated in each figure, the value of the GN^2 function will stabilize at the local minimum of the particular quadratic piece on which the variable knot is set. This stabilization occurs as a direct result of the knot movement in accordance with the centroid constraint described in Section B. However, we note that for most of the 'tiles', the local minimum occurs 'off' of the piece of the particular quadratic that specifies the GN^2 function value. This phenomena can be observed in each of the figures presented. When the phenomena does occur, it leads to a kind of 'cascading' effect in which the variable knot point automatically stabilizes itself at the smallest local minimum it can find along its cascading trail.

The true benefit of these graphs is derived from using them in conjunction with Table I, which depicts various knot movement scenarios. The table is divided into two halves, each outlining how the knots will move given initial guesses for their locations. The key feature to note, again, is how the knot location process occurs in two distinct stages: first, the assignment of the data points to the closest knot, and second, the determination of the centroid of the tiles. As described Section B of Chapter 4, ties are broken using a 'first-come-first-assigned' criterion, wherein the data point is assigned to the

first knot determined to be the closet. The knot location process ends when the same knot configuration occurs on two consecutive iterations.

For example, setting the knot points at 0.5 and -0.75, as seen in Figure 3.2, leads to the assignment of data points 0, 0.33, and 2 to the knot at 0.5, and data points -1, -0.5 to the knot at -0.75; hence, the first subset, {0,0.33,2} 'belongs' to 0.5, and the latter subset {-1,-0.5}, belongs to -0.75. The new knot locations are computed by summing the data points component-wise (there is only one component here) and dividing by the number of data points in that tile. Hence, in this example, the new knots are found at 0.78 and -0.75. The third iteration of the process yields the same result, thereby ending the search for the 'best' configuration; this is annotated in Table I as 'Stabilization'.

There is a complication, however, concerning the arbitrariness of the manner in which ties are broken. We note in the top half of Table I, where the 'first-come-first assigned' criterion was used in the knot assignment task, that the knots stabilized at -0.5 and 1.167. This is seen in Figure 3.4, where a local minimum can be observed at 1.167. But, as indicated in the same figure, having the variable knot stabilize at 1.167 does not yield the smallest GN^2 function value for this configuration of knot points: namely, -0.5, 1.167.

On the other hand, when the data points are assigned without use of the 'first-come-first-assigned' criterion, they stabilize at a different set of knot points, namely, -0.292 and 2. This case is depicted in the lower half of Table I, and can be seen in Figure 3.5. We note that Figure 3.5 contains the smallest GN^2 function value of all the knot

TABLE I

KNOT MOVEMENT SCENARIOS

Data Points located at: -1, -0.5, 0, 0.33, 2

Trial I

Iteration (Dscrptn)	Knot Pt $\hat{x}_1$	Data Point Assignment	Knot Pt $\hat{x}_2$	Data Point Assignment
0 Initial Guess	-0.75	{-1,-0.5}	0.5	{0,0.33,2}
1 New Knot	-0.75	{-1,-0.5,0}	0.78	{0.33,2}
2 New Knot	-0.5	{-1,-0.5,0}	1.167	{0.33,2}
3 New Knot	-0.5	{-1,-0.5,0}	1.167	{0.33,2}

S T A B I L I Z A T I O N

Trial II

Iteration (Dscrptn)	Knot Pt $\hat{x}_1$	Data Point Assignment	Knot Pt $\hat{x}_2$	Data Point Assignment
0 Initial Guess	-0.75	{-1,-0.5}	0.5	{0,0.33,2}
1 New Knot	-0.75	{-1,-0.5,0}	0.78	{0.33,2}
2 New Knot	-0.5	{-1,-0.5,0}	1.167	{0.33,2}
3 New Knot	-0.5	{-1,-0.5,0,0.33}	1.167	{2}
4 New Knot	-0.292	{-1,-0.5,0,0.33}	2.0	{2}
5 New Knot	-0.292	{-1,-0.5,0,0.33}	2.0	{2}

S T A B I L I Z A T I O N

point configurations considered, a value of 1.0. We can only conclude that we have not employed the criterion which will consistently yield the smallest GN^2 value, and note that this shortcoming warrants further investigation.

By following the development of the knot movement in the graphs of Figures 3.2 through 3.5, we can see how the best knots are found so as to minimize the GN^2 function value. The same process is followed in the 2K dimensional case, only it is not as easy to visualize.

E. OPTIMIZING THE KNOT LOCATION

The original problem of finding a set of knot points which minimizes the distance between the original data set and itself could be approached by minimizing the GN^2 value. However, such an approach does not necessarily provide an equitable representation by each of the knot points, which might be desirable. In minimizing the GN^2 value, there will be more data points per knot when the data is dense, than there will be when the data is sparse. We prefer to have the same, or nearly the same, number of data points per knot, which is what is meant by 'equitable representation.'

Furthermore, there is a distinct disadvantage in attempting to optimize using the GN^2 value, which has to do with accomodation of the tweaking process. During the development of the algorithm, we attempted to locate local minima of GN^2 by moving the knot points according to their relative weights in terms of their contributions to the overall GN^2 value. But, the region of interest was rife with local minima so that there was no positive assurance that attempting to balance the local norm contributions of each of the knots in this way would lead to

the objective of finding the global minimum norm. The absence of such a 'natural heuristic' for moving the knot points around the region of interest in search of the global minimum norm is clearly detrimental to the easy resolution of the problem, especially in light of the fact that optimization of the other quantity lends itself to such movement.

This leads to a different optimization constrained by the centroid specification as follows: minimize the sum of the squares of the number of data points in each tile minus the ratio of the data points to knot points, subject to the knot being the centroid of its tile. The function being optimized is a discrete function of 2K independent variables, whose domain is the positive integers (and may be zero in the case of C_j) and whose range is described by the expression

$$\sum_{j=1}^{K} (C_j - N/K)^2, \qquad (6)$$

where C_j is the count of the number of data points in the j^{th} tile. Further study of this function does not shed any more light on the minimization process and so it is abandoned.

By optimizing the DSUM value, the knot moving problem is facilitated; the obvious choice is to move knots with a high density of data points toward knots with a low density of data points. This 'shifting of the weights' approach proved successful, even though the value of GN^2 was no longer necessarily being minimized. (See Chapter 5, Table II for some specific results.) This tactic was further modified by making the knot movement 'symmetrical' so that the high density knots were moved toward the low density ones, and the low ones toward the high ones.

Thus, we wish to minimize the DSUM value expressed in Equation (6). We note that the DSUM value can vanish when each knot point represents

an equal number of data points, but since we have constrained the DSUM optimization somewhat, it will not usually vanish. In attempting this constrained optimization, we still seek a small GN^2 value (a local minimum value), but with the added feature that each of the knot points represent the same, or nearly the same, number of data points.

IV. DETAILS OF THE SUBROUTINES AND RELATED COMPUTER MATHEMATICS

A. INTRODUCTION

The general methodology of the KSLSTPS subroutine has already been summarized; this chapter explains the details of the subroutines written for the thesis and highlights the key aspects of each one. In designing an algorithm which minimizes the distance between two finite point sets, we assume that we are given the coordinates of the data points, the number of knots to be used to represent the data points, and an initial guess: the coordinates of some knot points.

Since the digital computer plays the role of workhorse in the knot selection process, the relevant mathematics performed by the computer is explained as well. As part of this discussion, we also consider the intracacies of solving an overdetermined system of equations and how certain contingencies are dealt with. This involves some additional interpolation and approximation theory, specifically Haar's theorem and its ramifications. We also offer some recommendations for implementing the subroutines on a microprocessor, and how one might initially decide on how many knot points to use.

B. MINORM SUBROUTINE

The MINORM subroutine basically performs two functions: first, it finds the centroid of the tiles; and second, it computes the value of DSUM for the particular knot point configuration at hand. We have already described how the minimization of the GN^2 value can be seen to

work in two steps: the determination of which data points 'belong' to the K knot points, followed by the centroid computation. This process is continued until a 'stable' knot configuration is attained.

In order to find which data points 'belong' to a particular knot, we use the smallest Euclidean distance criterion. For each data point, this distance from data point to knot is computed, followed by assignment of the data point to the appropriate (closest) knot. If ties are present, they are broken in the sense that the data point is assigned to the first knot determined to be the closest. This results from the sequential, nested design of the algorithm in which a data point is compared to each knot point and assigned at the completion of the comparison before the next data point is considered.

As a data point is compared to each of the K knots, it is assigned to the closest knot and its x and y components are separately added to two zero vectors, each consisting of K elements; thus, a 'running' total of the sum of the x-components and y-components for each of the knots is sequentially maintained, based on which particular knot is the closest to that data point. Then, as we have seen, the new components of each of the knot coordinates are found by dividing the elements of the two vectors of sums by the number of data points belonging to the knot points.

The subroutine MINORM has another noteworthy feature. It concerns the case wherein a knot point is so far removed from the data points, that none of the data points is assigned to it during the knot assignment procedure previously described. When this occurs, the knot

is moved so that its location coincides with that of the nearest data point; at the next iteration, there is at least one data point belonging to it.

As we have seen, the computation of the quantities global norm (GN^2), and global tile difference (DSUM) are important since the optimization occurs with respect to the second constrained by the first. Both of these values will also play a key role in the subroutine for moving knot points around the region of interest. Additionally, this subroutine counts the number of data points which 'belong' to each knot point, and calculates the individual contributions of each of the tiles to the overall GN^2 value. These quantities are labeled NI and LN, respectively, and are passed as output parameters by the subroutine.

Heuristically, we have found that there is no positively 'sure' way to ascertain either the value or the location of the minimum GN^2 value. However, the scheme developed here seems to be fairly consistent in obtaining 'good' sets of knots, but does not necessarily obtain the same set of knots when different starting configurations are used. This is a little bothersome, but not totally unexpected. We also note that the GN^2 value is not well correlated to the DSUM value. Furthermore, this observation is not unexpected and does not diminish the success achieved with the subroutine design.

C. TWEEK SUBROUTINE

The objective of the TWEEKing subroutine is to move the knots around the plane, so that the boundaries of the Dirichlet Tesselation change in order to seek a minimum value for DSUM. The basic idea in the TWEEKing process is to move one knot at a time along a straight line segment by

various distances; the line connects the knot with the most data points to the knot with the fewest data points. The TWEEKing subroutine first calls upon the MINORM subroutine to center the initial configuration of the knots in their tiles.

As pointed out Section B, MINORM also computes initial values of GN^2 and DSUM; the first action by the subroutine TWEEK is to call the MINORM subroutine and obtain these initial parameters. These values are then used to initialize the variables GNSAV and DSMSV which are to be used in determining the best knot configuration. As better knot configurations are found in the TWEEKing process, they will be saved along with their corresponding GN^2 and DSUM values, as the variables GNSAV and DSMSV. Hence, each time a new configuration is tested, the values of GNSAV and DSMSV will be involved and updated only when a better result is achieved.

The number of data points associated with each of the knot points has already been determined in MINORM, so that, based on this information, the knot points having the most data points and those having the fewest data points are singled out and indexed for future reference. If the number of points per tile is 'nearly equal,' the subroutine quits and returns the coordinates of the knot points corresponding to this configuration. Based on this information, every possible combination of knots with the most points to knots with the fewest points is made, which we shall refer to as 'the most-to-fewest pairs'.

For a given configuration of knots, a 'system' of lines can be envisioned connecting the most-to-fewest pairs along which new configurations of knot will be generated. Along each of these lines, we

first move the knot with the fewest points toward the knot with the most points, keeping the latter fixed, and invoke the subroutine MINORM at each 'stop' along the way to ascertain whether or not a better result is achieved. This is immediately followed by movement of the knot with the most points toward the knot with the fewest, keeping the latter fixed, after which the subroutine MINORM is again invoked; each of these moves is termed a tweak. Thus, a 'symmetric' avenue of approach is taken in the TWEEKing process.

The distance along the straight line of each tweak is determined by a pre-set parameter called DIVisor which varies as a function of the index, p, which runs from 1 to 10. The parameter raises 1.5 to the p^{th} power and scales this number by a factor of 0.8, so that along any given line in the system a 'stop' is made at 83, 56, 37, 25, 17, 11, 7.3, 5, 3.3, and 2.2% of the total initial distance between the knot points (assuming the index p is exhausted).

The subroutine MINORM is called each time a different configuration of knot points is 'proposed' as a result of a tweak, and the new values of GN^2 and DSUM, which are passed back to the TWEEK subroutine are used in subsequent tests to ascertain whether the configuration of knot points in question has produced the best results to date. In determining whether or not a particular configuration of knot points is better (and should be saved for future reference and comparison), tests are made and the results saved according to the general criteria below.

1) Whenever a smaller DSUM value is found (as compared to its value based on the iteration's initial configuration for the knot points), a second test is done to compare this value to the value of DSMSV; when DSUM is found to be smaller, this configuration of knot points is saved and the values of DSMSV and GNSAV are updated.

2) In the event that a tie exists between the present DSUM value and its value based on the iteration's initial configuration for the knot points, the tie is broken using the present GN^2 value as compared to its value based on the iteration's initial configuration for the knot points. When the tie is broken in favor of the present GN^2 value, a second test is made to compare this value of GN^2 to the value of GNSAV. As before, when the test proves positive, this configuration of knot points is saved and the values of GNSAV and DSMSV are updated.

3) When the values of DSUM and GN^2 returned by subroutine MINORM are equal to those values found as a result of the iteration's initial configuration for the knot points, an assumption that the present configuration of knot points is the same as those resulting from the iteration's initial configuration is made. When this occurs, the coordinates of the knot points are set back to their starting values (those resulting from the iteration's initial configuration), and the next pair of knots is tweaked as described above.

4) It is conceivable for all of the possible (twenty total) stops along the straight line to be checked (alternating moves of the knot with the fewest data points toward the knot with the most of, and the knot with the most toward the knot with the fewest). When (and if) all these stops (calls to the MINORM subroutine) have been checked, the next pair of knots (one with the most data points and the other with the fewest) is considered in the same way.

We note that as soon as a better result is found along the straight line, the next pair of knots is considered, and the rest of the stops along the straight line are never checked. However, we also note that every possible combination of knots with the most data points with knots with the fewest data points is considered before the TWEEK subroutine returns its output parameters: the configuration of knot points which minimized the DSUM value subject to the constraint that each knot point be located at the centroid of its tile.

After an iteration of the TWEEK subroutine, the new candidate set for best configuration of knot points is again subjected to the 'most-fewest data point determination,' followed by the procedure outlined

above. Only when the same best configuration is arrived at does the subroutine return its output parameters.

D. KNTIG SUBROUTINE

We have specified the input for the KSLSTPS package in general terms to be the following information: the coordinates of the N data points, the number of knots to be used, and the coordinates of these knots. In fact, the KSLSTPS package must also be provided with information such as the number of data points, the value of the dependent variable at each of the data points, the number of intervals in each of the x and y directions on the evaluation grid, and the interval lengths of the grid in both the x and y directions. The KNTIG subroutine internally generates an initial guess for the knots based on the number of knots the user specifies, and some of the additional information listed above.

There are obviously many different ways in which an initial guess for the knots can be made. For example, they could be generated randomly, or based on a certain probabilty density function, or positioned so that they cover the region of interest in some uniform fashion. Alternatively, we could position the knot points by 'eye-balling' the data. Each of these methods has its own merits, but we selected the uniform approach because it was the easiest to automate, and it seemed to have the smallest potential for complications. Spacing the knot points uniformly can also be accomplished in several different ways, and the method described here can certainly be modified to accomodate a user's needs.

Since we are attempting to minimize the DSUM value subject to the constraint that each knot be the centroid of its tile, it appears that such an approach may not be optimal in terms of leading to the fastest resolution of the knot selection process. However, we noted earlier that there is no way to ascertain the locations of the knots leading to the smallest GN^2 value (or the value of the minimum GN^2 itself). Thus, this shortcoming may be overlooked and may even be an advantage, since we may assume that a bigger search is a better (more exhaustive) search, within reasonable limits, of course. We shall define a reasonable search to be a 'thorough' search in the 'right' area, a description that can be applied to this method. We also note that in most cases, different initial guesses will cause the knot selection process to follow different searching patterns, which could potentially lead to a better solution.

The best number of knot points to be used for a given set of data will depend on the user's needs. Obviously, the use of several different initial guesses, along with several different numbers of knots will lead to what may heuristically prove to be the best set of knot points. Factors such as the number of data points and the time available will also play a role in this decision. As a place to start, we recommend using the square root of the number of data points as the number of knots, all other factors considered being equal. When some factors require more consideration than others, this initial number may be significantly or slightly modified, depending on the subjective judgement of the user.

Once the number of knots has been specified, the KNTIG subroutine will calculate a number of horizontal lines (NLINES) to be 'drawn' in the region of interest, which is also specified through the input parameters of the subroutine. The NLINES variable is the square root of the number of knots points specified, rounded to yield an integer. This is followed by a determination of the number of knots per line, based on an even distribution of the knot points. When an equal distribution of knot points per line is not possible, the extra knots are distributed evenly between the middle lines, and such that the knots along a given line are evenly spaced.

E. GEVGRD SUBROUTINE

The GEVGRD subroutine uses the newly found coefficient vector (to be discussed in the next section) to evaluate the newly constructed approximating function F at each point specified on the grid, which is also described by the additional information above.

The objective of the GEVGRD subroutine is to provide a rectangular grid of points on which the newly constructed approximating function can be evaluated, and to perform the evaluation. The subroutine outputs the values of the approximated surface which can be used to obtain a three dimensional plot of the surface, or as input to a product approximation for the surface. The procedure is straight forward and relies on the coefficient vector containing A_j, j=1,...,K, and a_0, a_1, and a_2 found in solving the overdetermined system of linear equations, described in the next section. With this vector, and using the grid points as the new data points $(\bar{x}_i,\bar{y}_i)$, i=1,...,$g_x \cdot g_y$, (where g_x is the number of points on the grid in the x direction, and g_y is the number of points in the y

direction), and the optimized knot points, $(\hat{x}_j, \hat{y}_j)$, $j=1,\ldots,K$, the value of the function F is found at each of the grid points.

This involves taking the inner product of the coefficient vector with the linear combination of the basis functions evaluated at the knot and grid points. For each grid point, there will be an equation of the form

$$F(\bar{x}_i, \bar{y}_i) = \sum_{j=1}^{K} A_i d_{ij}^2 \cdot \log(d_{ij}) + a_1 \bar{x}_i + a_2 \bar{y}_i + a_0,$$

where

$$d_{ij} = (\bar{x}_i - \hat{x}_j)^2 + (\bar{y}_i - \hat{y}_j)^2,$$

so that a dependent value of the approximated surface is found for each point in the grid.

F. LSCOEF SUBROUTINE

1. QR Decomposition

We wish to compute the coefficients, A_j, $j=1,\ldots, K$, a_0, a_1, and a_2, of the least squares thin plate spline

$$F(x,y) = \sum_{j=1}^{K} A_{ij} d_{ij}^2 \cdot \log(d_{ij}) + a_1 x + a_2 y + a_0,$$

where

$$d_{ij}^2 = (x_i - \hat{x}_j)^2 + (y_i - \hat{y}_j)^2,$$

so that the minimization over these coefficients of the residuals is achieved. As was previously mentioned, this leads to an overdetermined system of linear equations which can be written in matrix notation as

the following, where the dimensions of each of the blocks in the matrix are indicated.

$$
\begin{array}{c}
\\
N \\
\\
\\
\\
D^{-1} \\
\\
3 \\
\\
\\
\end{array}
\begin{bmatrix}
d_{11}^2 \cdot \log(d_{11}) & d_{12}^2 \cdot \log(d_{12}) & \cdots & d_{1K}^2 \cdot \log(d_{1K}) & x_1 y_2 1 \\
d_{21}^2 \cdot \log(d_{21}) & d^2{}_{22} \cdot \log(d_{22}) & \cdots & d_2^2 \cdot \log(d_{2K}) & x_2 y_2 1 \\
\cdot & \cdot & & \cdot & \cdots \\
\cdot & \cdot & & \cdot & \cdots \\
\cdot & \cdot & & \cdot & \cdots \\
d_{N1}^2 \cdot \log(d_{N1}) & d_{N2}^2 \cdot \log(d_{N2}) & \cdots & d_{NK}^2 \cdot \log(d_{NK}) & x_N y_N 1 \\
\hat{x}_1 & \hat{x}_2 & \cdots & \hat{x}_K & 0\ 0\ 0 \\
\hat{y}_1 & \hat{y}_2 & \cdots & \hat{y}_K & 0\ 0\ 0 \\
1 & 1 & \cdots & 1 & 0\ 0\ 0
\end{bmatrix}
\begin{array}{c}
K \\
\\
\\
\\
\\
\\
3 \\
\\
\\
\end{array}
\begin{bmatrix}
A_1 \\ A_2 \\ \cdot \\ \cdot \\ \cdot \\ A_K \\ a_1 \\ a_2 \\ a_0
\end{bmatrix}
\begin{array}{c}
\\
K \\
\\
\\
\\
\\
\\
3 \\
\\
\\
\end{array}
= D^{-1}
\begin{array}{c}
\\
N \\
\\
\\
\\
\\
\\
3 \\
\\
\\
\end{array}
\begin{bmatrix}
f_1 \\ f_2 \\ \cdot \\ \cdot \\ \cdot \\ f_N \\ 0 \\ 0 \\ 0
\end{bmatrix}
$$

$$
\begin{array}{c}
\\
1 \\
\\
\\
\\
\\
\\
1 \\
\\
\\
\end{array}
\qquad
\begin{array}{c}
\\
1 \\
\\
\\
\\
\\
\\
1 \\
\\
\\
\end{array}
$$

Here, the matrix D is an N+3 by N+3 diagonal matrix containing the standard deviations of error at the i^{th} data point for each of the N data points, and the inverse weights associated with the three equilibrium equations. Thus, D may be expressed in matrix notation as

$$
\begin{bmatrix}
\sigma_1 & & & & & & & & \\
& \sigma_2 & & & & & & 0 & \\
& & \ddots & & & & & & \\
& & & & \sigma_N & & & & \\
& 0 & & & & 1/w_1 & & & \\
& & & & & & 1/w_2 & & \\
& & & & & & & 1/w_3 &
\end{bmatrix}
$$

The LSCOEF subroutine solves for these coefficients using the optimized knot points, and the given set of data points including the dependent variable, f_i, on the right hand side. As we just saw in the previous

section, once these coefficients have been computed, they are used in evaluating the surface at points specified on a rectangular grid of points.

We note that the three additional equations in this system, corresponding to the equilibrium conditions described in Chapter 2, can be treated in one of two ways. First, we could solve them in the least squares sense as part of the larger least squares problem, or second, they could be treated as highly accurate constraints to be satisfied exactly [Ref. 15:pp. 148-149]. As constraints, the equilibrium conditions can be interpreted to specify that the approximation surface (i.e., the least squares TPS based on the optimized knot point locations) be rigid and not subject to any composite rotational or translational forces. As was the case in TPS interpolation where no translation or rotation occurred, we wish to have the least squares TPS approximation subjected to the same constraints.

One way to impose these equations as constraints in the TPS approximation is to heavily weight the three equilibrium conditions (in relation to the other equations in the system). The net effect of such a ploy is to solve these equilibrium equations 'exactly' while the other equations in the system are solved in the least squares sense. In our case, the equilibrium equations are weighted by a factor of 1000.

The method chosen to solve this overdetermined system of linear equations $A\bar{x} = b$ is the QR decomposition, where the original coefficient matrix is factored into an orthogonal matrix Q times an upper triangular matrix R. [Ref. 6:p. 131]. Alternatively, the least squares problem could be solved using the normal equations or the Singular Value

Decomposition (SVD). However, the QR decomposition displays a great deal of numerical stability, which is characteristic of orthogonal (Householder) transformations. It also has the added advantage of being adaptable to special requirements, such as the sequential modification of the matrices corresponding to the accumulation of more data (to be discussed later). We note that orthogonal transformation matrices have a natural place in least squares computations because Euclidean lengths are preserved in multiplication. [Ref. 15:pp. 4-22]

In theory, once we have performed the factorization $A = QR$, it is easy to solve the least squares problem $A\bar{x} = b$. Substituting for A, we have

$$QR\bar{x} = b$$

and multiplication by Q^T yields

$$R\bar{x} = Q^T b,$$

since $Q^T Q = I$. R is a rectangular matrix of size N+3 by K+3, which is zero below its main diagonal, so that we have

$$\begin{matrix} K+3 \\ N-K \end{matrix} \begin{bmatrix} R_{11} \\ 0 \end{bmatrix}_{K+3} \bar{x} = \begin{bmatrix} Q^T b_1 \\ Q^T b_2 \end{bmatrix}_{1} \begin{matrix} K+3 \\ N-K \end{matrix}$$

in block matrix notation. Thus, $\bar{x} = R_{11}^{-1} Q^T b_1$ and the 2-norm of the residuals is $||Q^T b_2||_2$.

In summary, the computation of x requires only the matrix-vector multiplication $Q^T b$, followed by back substitution in the triangular

system $R\bar{x} = Q^T b$. Using a Householder algorithm for the QR decomposition, numerical stability is guaranteed. [Ref. 6:p. 131]

2. Rank Deficiency and Haar's Theorem

The LINPACK subroutines, SQRDC and SQRSL [Ref. 16:pp. 9.1-9.15], are instrumental in first setting up the QR decomposition, and then in manipulating it to compute the least squares solutions. Provision is made in the event that the coefficient matrix A is rank deficient (i.e., it has less than K+3 linearly independent columns). However, it is highly unlikely that a rank deficient coefficient matrix will arise. We note that as part of the calling sequence for the subroutine SQRSL, the rank of the coefficient matrix is specified to be full, thereby reflecting the confidence in the following logic. We could attempt to prove this statement using the idea of unisolvence and Haar's theorem. [Ref. 1:pp. 31-32]. Ultimately, such an attempt will fail as we shall examine why. Alternatively, we can argue against the occurrence of a rank deficient coefficient matrix from a much simpler standpoint.

We begin by constructing a surface which may lead to a rank deficient coefficient matrix. Choose a nontrivial thin plate spline surface with a set of given knot points, which extends across the x-y plane (i.e., it possesses both positive and negative values). Then, out of the infinite number of points with value zero at the surface, we choose a finite number of points, and call them our data points. Now, given this finite set of points, one of the possible interpolating surfaces which can be constructed to interpolate these points is the surface described by $F(x,y) = 0$. Obviously, the coefficient matrix for this surface will be rank deficient, since we know at least two

solutions to the interpolation problem exist, the original function and the zero function.

The question now becomes: for such a surface, what are the chances that, for a given set of data, a configuration of knot points (obtained via the knot selection process incorporated in the subroutine MINORM and TWEEK), will occur leading to a rank deficient coefficient matrix, as in the case described above? Statistically, we must conclude that this situation is nearly impossible and thus, the singular system of equations or rank deficient coefficient matrix will be unlikely to occur.

In the event that the situation could and did arise, however, the subroutines SQRDC and SQRSL can resort to the use of a 'truncated' QR decomposition, in which the least squares problem is solved using fewer than the given number of basis functions. In other words, since the coefficient matrix is rank deficient, two or more of the columns are linearly dependent and, hence, their 'copies' are not used in the solution of the system. This is tantamount to eliminating one or more of the basis functions, since it (they) can be looked at as a linear combination of the other basis functions on the given set of data points. Alternatively, the SVD method could be employed to solve the rank deficient problem in which the solution minimizes the 2-norm of the residuals.

When the coefficient matrix A is rank deficient, the back substitution process used to compute the least squares solution breaks down. However, by permuting the columns of the coefficient matrix so that R becomes

$$\begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix} \begin{matrix} r \\ m-r \end{matrix}$$

$$\begin{matrix} r & n-r \end{matrix}$$

where R_{11} is upper triangular, the least squares problem is readily solved as follows. After the computation of $Rx = Q^T b$, where $x = \pi(y\ z)^T$, π is the permutation matrix, and $Q^T b = (c\ d)^T$, we have

$$\begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix} \pi \begin{bmatrix} y \\ z \end{bmatrix} = \begin{bmatrix} c \\ d \end{bmatrix}$$

so that

$$\bar{x} = \pi \begin{bmatrix} R^{-1}(c - R_{12}z) \\ z \end{bmatrix}.$$

When z is set to zero, we obtain the basic solution

$$x_B = \pi \begin{bmatrix} R_{11}^{-1} c \\ 0 \end{bmatrix}.$$

However, we note that the solution does not necessarily minimize the 2-norm of the residuals $||Ax - b||_2$ unless the submatrix R_{12} is zero also. [Ref. 17:pp. 162-163]

We now mention an unsuccessful attempt we made at extending the idea of unisolvence and employing Haar's theorem to the TPS interpolation (not approximation) problem. Suppose we have the functions $f_1(x), f_2(x),\ldots,f_n(x)$ defined in one dimension (the interval I), and we are given n distinct points $x_1, x_2,\ldots,x_n \in I$, each corresponding to a value w_j, $j=1,\ldots,n$. Then it is a known fact that the interpolation problem

$$\sum_i a_i f_i(x_j) = w_j,$$

can be solved uniquely if and only if the determinant $|f_i(x_j)| \neq 0$. [Ref. 1:p. 31]

Such a system of n functions defined on a point set S is called unisolvent on S, if the requirement on the determinant above holds for all n distinct points lying in S. [Ref. 1:p. 31] Thus, unisolvence implies that a unique solution exists which in turn means that the coefficient matrix has full rank. This follows from the fact that a homogenous system of equations has only the trivial solution if and only if the coefficient matrix has full rank.

Unisolvent systems are reasonably abundant in one dimension, but, as in our particular case where the point set is contained in two dimensional Euclidean space, Haar's theorem asserts that the system of functions cannot be unisolvent. In proving Haar's theorem [Ref. 1:p. 32], Davis' ploy is to interchange the locations of two points wherein the determinant changes sign, inferring that at some point in the switching process, an intermediate position of the points was attained at which the determinant vanished. This interchanging of the locations of two points is tantamount to interchanging two rows in the coefficient matrix.

However, it turns out that Haar's theorem cannot be applied to the TPS interpolation problem and it is even less appropriate for the TPS approximation problem. In the interpolation problem, the basis functions vary with the data points, so that by interchanging the locations of the two data points, we not only interchange two rows in the coefficient matrix, we also interchange two columns, corresponding

to the basis functions varying with the data points. In essence, we end up with the original coefficient matrix, with two rows and the corresponding columns each interchanged, and thus, the original determinant is reproduced as well. We conclude then that Haar's theorem cannot be applied to the TPS interpolation problem, and next we shall see why it does not apply to the TPS approximation problem either.

In the TPS approximation problem, we have an overdetermined system of linear equations, so that the coefficient matrix is rectangular with the number of rows being greater than the number of columns. If the knot points are chosen without regard to the data points, it seems plausible to have more than one solution. This follows from consideration of the two dimensional space in which we wish to fit a plane given three arbitrarily chosen points. If these three knot points are collinear, then any plane containing this line satisfies the system and we will have a rank deficient coefficient matrix. The same argument can be made in the TPS approximation problem, except for the fact that we are picking the knot points as they relate to the nearest data points (i.e., based on the criterion that each knot point be the centroid of the data points in its tile). Since we are choosing the knot points in conjunction with the data points, this argument is not valid for the TPS approximation problem.

We suspect that locating the knot points in this way guarantees full rank in the coefficient matrix. But even though several similarities exist between the proof of Haar's theorem and the TPS approximation problem, Haar's theorem cannot be used to confirm our suspicions for the following reason. As was previously mentioned, the proof of Haar's

theorem in interpolation rests on being able to interchange two points, thereby inducing a sign change amongst the respective determinants.

In TPS approximation, this procedure amounts to fixing the knots and considering the movement or interchanging of two data points within a single tile. The data points must be switched without disturbing the location of the centroid of the tile and without having the two points coincide during the switch. One way this may be accomplished is to define an ellipse with vertices at each of the two data points. The ellipse must be contained inside the tile; if one data point is on the boundary of the tile, then the other data point is chosen so that it is not on the boundary of the tile. Then, moving at equal rates along the ellipse in the same (say clockwise) direction, the switch can be accomplished meeting the criteria established above.

The complication arises from the fact that all possible K+3 order determinants must vanish simultaneously in the same way in which the one determinant vanished during the point interchanging scheme in the proof of Haar's theorem in interpolation. It is simply not clear how and if this can be accomplished in the overdetermined system of linear equations found in the TPS approximation problem. We must conclude that the employment of a proof similar to that for Haar's theorem again eludes us.

Another contingency we wish to address concerns a potential user who lacks the requisite amount of computer storage space (e.g., he is confined to use of a microprocessor). Under such circumstances, updating of the QR decomposition can be employed to overcome this resource deficiency. Using the LINPACK subroutines SQRDC and SQRSL, an N+3 by

K+3 coefficient matrix A must be stored so that the QR decomposition can be made. For large systems, we see a great need for additional storage to accomodate the large coefficient matrix along with some others such as the solution and residual vectors which result.

Since there is no requirement to decompose the entire coefficient matrix and solve for the least squares solution all at the same time, we could solve the system in steps, starting with the first $K+3 < I < N+3$ equations, where I depends on the amount of storage available, but must be greater than K+3 in order to have an overdetermined system. The QR decomposition would be applied to this I by K+3 coefficient matrix. This is followed by the appending of a block of up to N+3-I more equations to the already resolved system's coefficient matrix block, which now has zeroes below the diagonal, except in the appended block. By updating the system in this way, the problem is reduced in dimension and solved in steps until it is completed, a relatively simple proposition in light of the Householder and other appropriate orthogonal transformations available. Hence, a significant advantage is achieved in terms of reducing the storage space requirements for solution of a large system of equations, since additional space is not needed to accomodate the large coefficient matrix all at the same time.

V. EXPERIMENTAL RESULTS AND CONCLUSIONS

A. INTRODUCTION

The overall objective of this thesis was achieved in terms of the creation of a method for representing a large data set with a significantly smaller one with which to approximate surfaces, and a set of FORTRAN subroutines to implement it. To support this claim, we present an overview of the data sets used throughout the experimentation to develop the algorithm, as well as a description of the various large data sets which were used to test and verify it. Then, we discuss how these data were used and the results attained in their appropriate contexts.

First, we show how the data sets used in the experimentation evolved; then, we present the results obtained in attempting to represent a large set of data with a significantly smaller set of knot points. We include a table of the numerical values of global norm (GN^2) and global tile difference (DSUM) found in using the program to optimize the knot point locations as described in Chapter 3. We also portray the data sets graphically, and illustrate some of the optimized knot point configurations found in the experimentation.

Next, we turn to the underlying surface and look at the closeness of fit between the constructed surface F and the true surface; this is done in the context of three measurements of error: the maximum, mean, and root-mean-squared (RMS) errors on the residuals and on the grid. We present some surfaces derived by other means for 'comparison' (albeit between apples and

oranges), and summarize the results for the data sets described in Section B in terms of the most meaningful measure of the error: the RMS values.

Finally, we look at the experiments in the context of time with the idea of identifying a more efficient approach to take. We also discuss one ramification of not meeting the underlying assumption.

B. REPRESENTING THE DATA

Initially, in deciding how to best resolve the problem, we experimented with small sets of 25, 33, 50, and 100 data points and 5, 6, 7, and 10 knot point sets. These data sets had been used extensively in previous surface modeling work by Richard Franke and others and provided a well established, solid standard from which to embark. The data sets can be characterized as random distributions of data points on a unit square, although each of them was derived from a surface generated by some known function. The knot point numbers were derived from the square root of the number of data points criterion previously mentioned. Locations for the knot points were generated by 'eyeballing' the data, randomly, and ultimately, by distributing them uniformly throughout the region of interest.

Throughout the initial phase of experimentation, our goal was to locate the knot points which minimized the GN^2 value. However, it became readily apparent at the outset that there was *no* definite way to ascertain that we had found the minimum GN^2 value, because results of varying quality were achieved under various conditions, such as different initial guesses for the knots and the resulting different search patterns taken. The erratic behavior of the final GN^2 values as contrasted with the initial and minimum GN^2 values for four of the surfaces used can be observed in Table II. More importantly, as was pointed out in Chapter 3, optimization of DSUM lends

TABLE II

MINIMA ACHIEVED IN TESTING

Data Set (Descrptn)	Number of Knot Pts	Initial $GN^2/DSUM$	Minimum $GN^2/DSUM$	Final $GN^2/DSUM$
100 (Original Random Surface)	10	1.99/24.0	1.98/4.0	1.99/4.0
	20	0.90/36.0	0.86/18.0	0.87/10.0
	25	0.69/28.0	0.61/4.0	0.61/4.0
200 (Cliff)	20	1.69/146.0	1.69/146.0	1.81/16.0
	25	1.31/286.0	1.27/114.0	1.35/36.0
	35	0.87/239.1	0.80/49.14	0.88/23.1
200 (Humps & Dips)	20	1.69/90.0	1.60/46.0	1.63/22.0
	25	1.31/88.0	1.27/40.0	1.30/26.0
	35	0.89/83.1	0.85/31.14	0.86/29.1
500 (Humps & Dips)	25	2.95/1280.0	2.78/424.0	2.85/364.0
	50	1.44/1190.0	1.35/410.0	1.38/354.0

itself to the knot moving algorithm employed in the TWEEK subroutine since a 'natural heuristic' is present in the quantity. This lead to the decision to subjugate the goal of locating the minimum GN^2 value and pursue the minimum DSUM value under the constraint that each knot point be the centroid of its tile. In this way, we essentially acknowledged the obvious outcome of attempting a finite search over a multi-dimensional set, and recognized that it could not be accomplished efficiently.

The importance of the minimization of the DSUM value cannot be overemphasized in contrast to the minimization of the GN^2 value; theoretically, it may not be possible to accomplish the minimization of DSUM in a unique way,

but given the constraint above, the problem is defined sufficiently to effect a reasonable outcome. Furthermore, under the assumption that the data density reflects something about the behavior of the overlying surface, the minimization of the DSUM value takes on added significance, since we now have a situation wherein the density of the knot points is indicative of the underlying surface as well.

Up to this point, we have been concerned about the independent variables of the underlying surface in the test data generation process, but this approach was modified somewhat in continuing the investigation. Since we had begun to achieve reasonable results with the smaller data sets, we moved up to larger data sets of 100, 200, and 500 data points each, and knot point sets of 20, 25, 35, and 50 knot points each.

The large sets of data, including the 100 point 'original' set, the 200 point 'humps & dips' set, the 200 point 'cliff' set, the 500 point 'humps & dips' set, and the 1669 point 'hydrographic' set are shown in the odd numbered Figures 5.1 through 5.9. These larger data sets (except for the last) were generated using known functions in a way which forced the disposition of points to be proportional to the curvature of the sampled function. The basic idea behind creating these data sets was to divide the region of interest into some number of squares, followed by the computation of an 'average' value of curvature at the centroid of each square; then an appropriate number of data points are positioned randomly within the square based on the average curvature value, thereby reflecting the curvature in the density of the data. This was done for 100 points based on some known functions, and these were then augmented with the original 100 point set

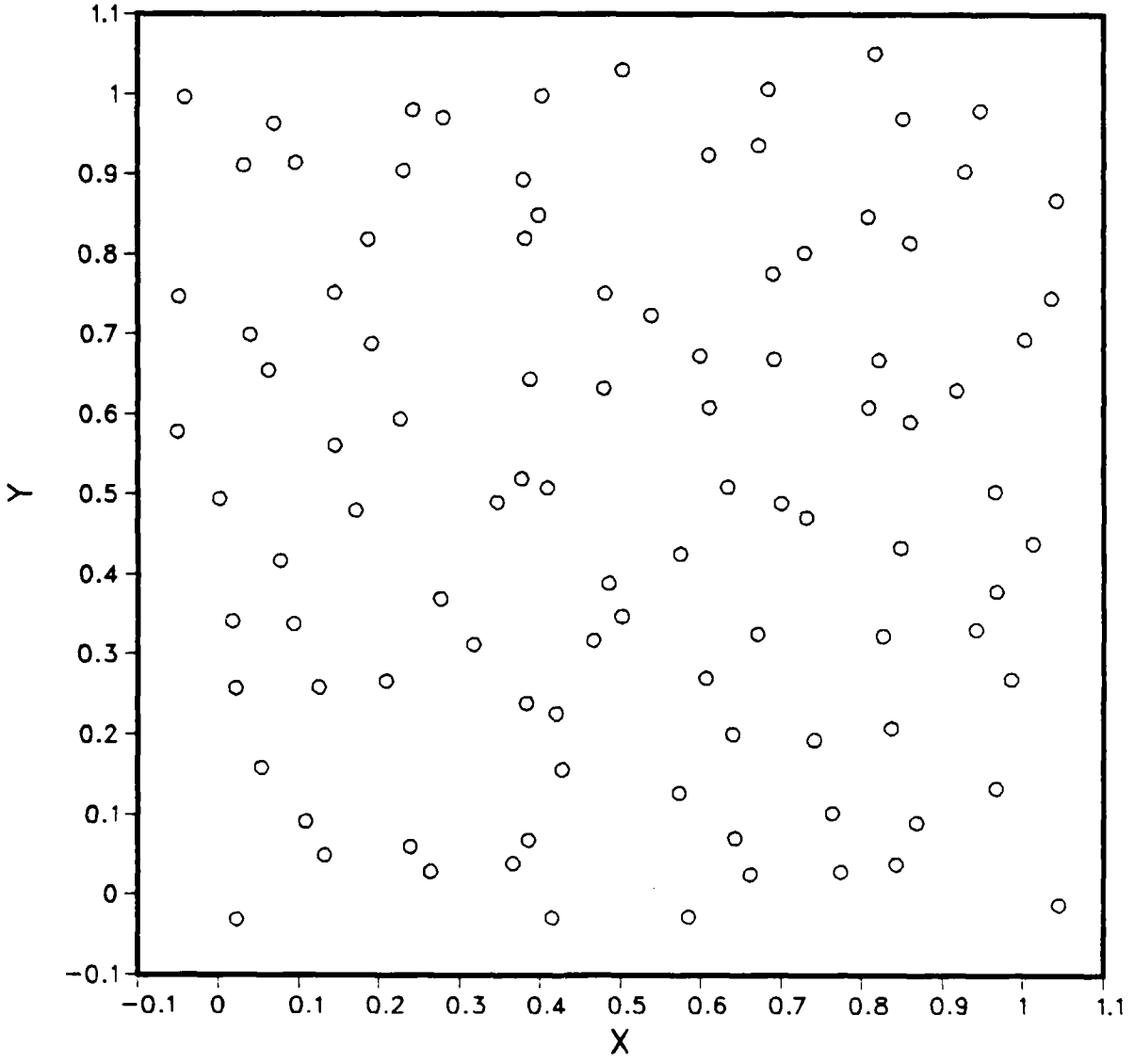


Figure 5.1. 100 Point Original Data Set.

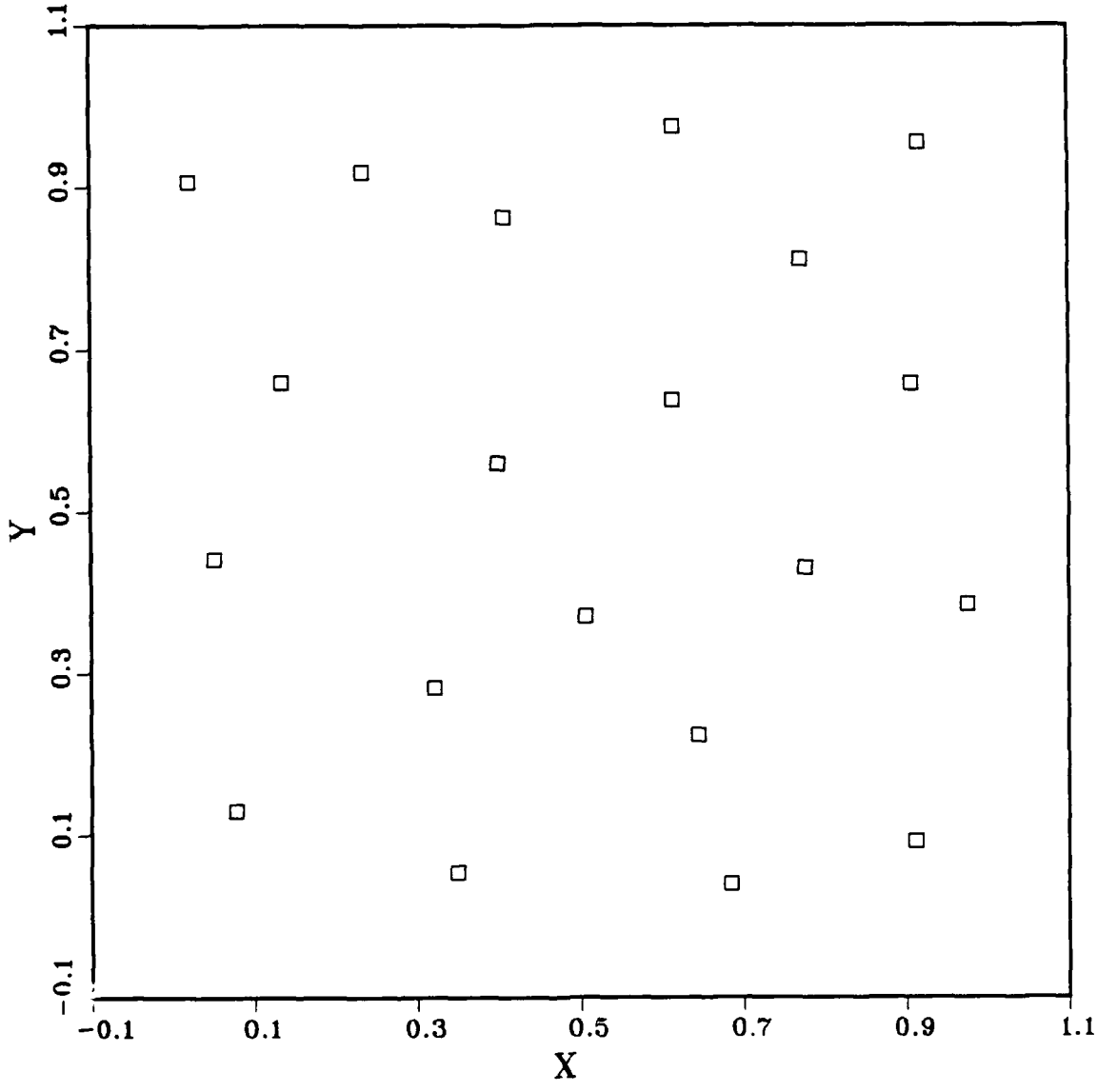


Figure 5.2. 20 Knots Representing Original 100 Point Set.

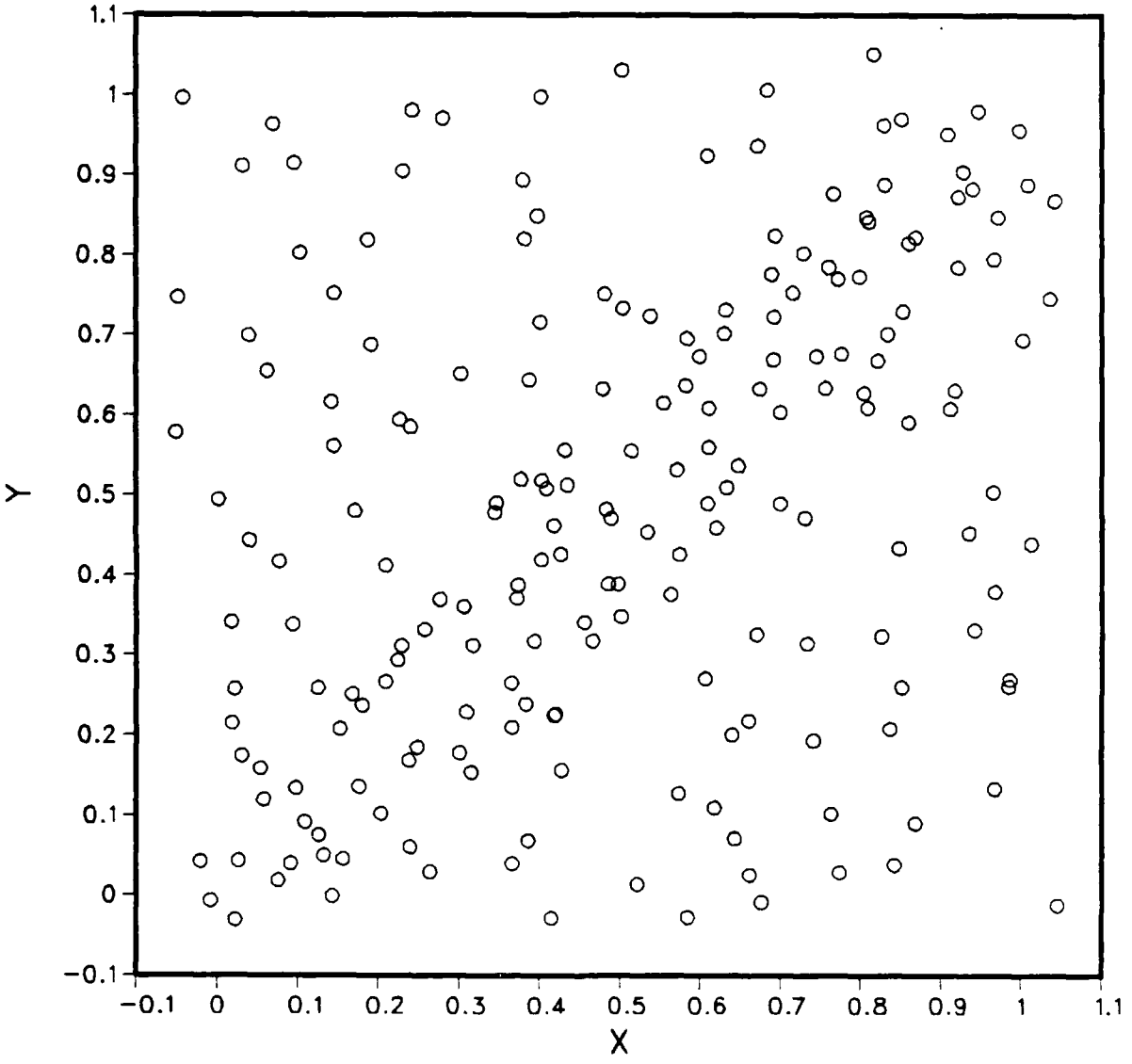


Figure 5.3. 200 Point Cliff Data Set.

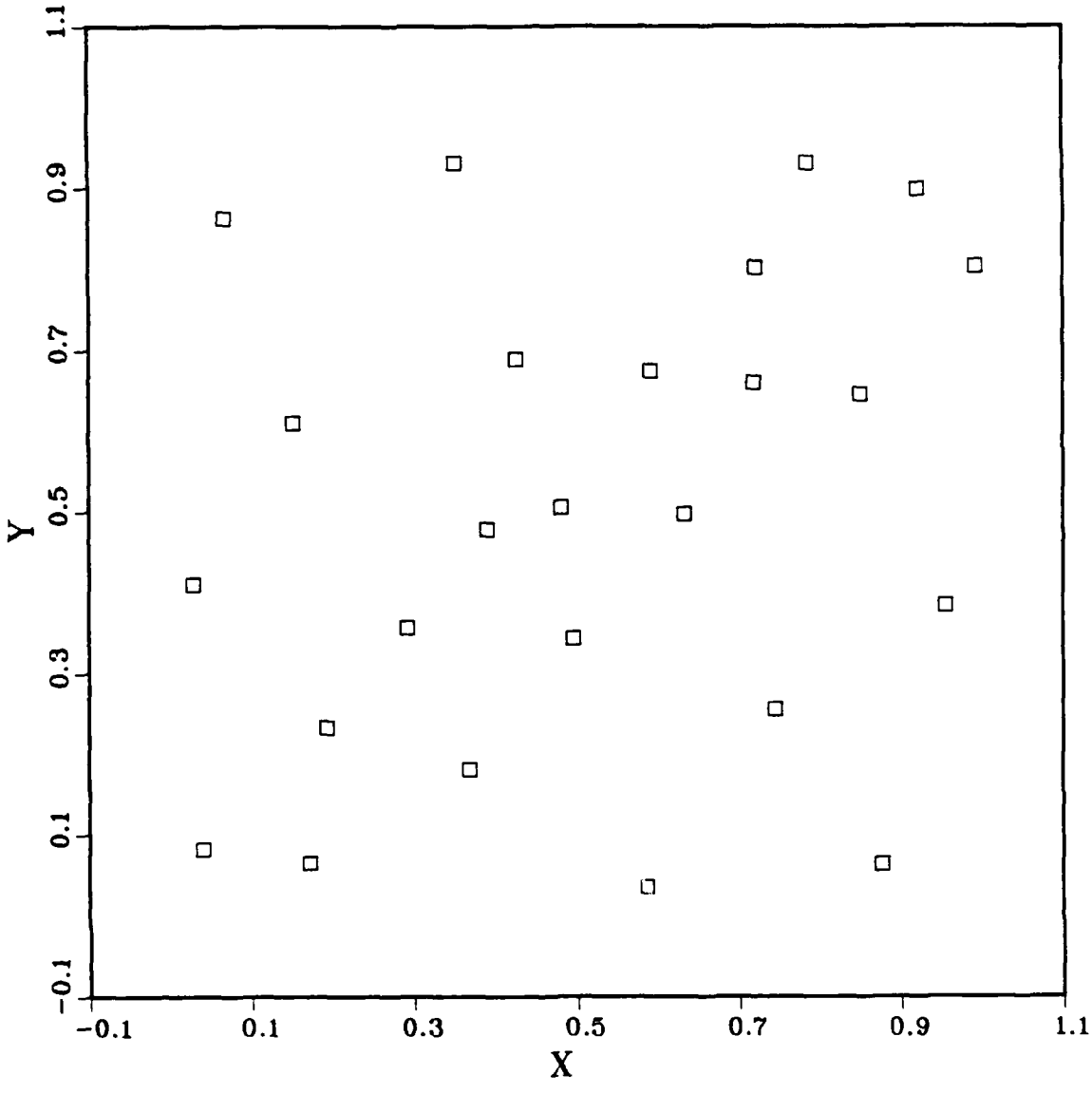


Figure 5.4. 25 Knots Representing 200 Point Cliff Set.

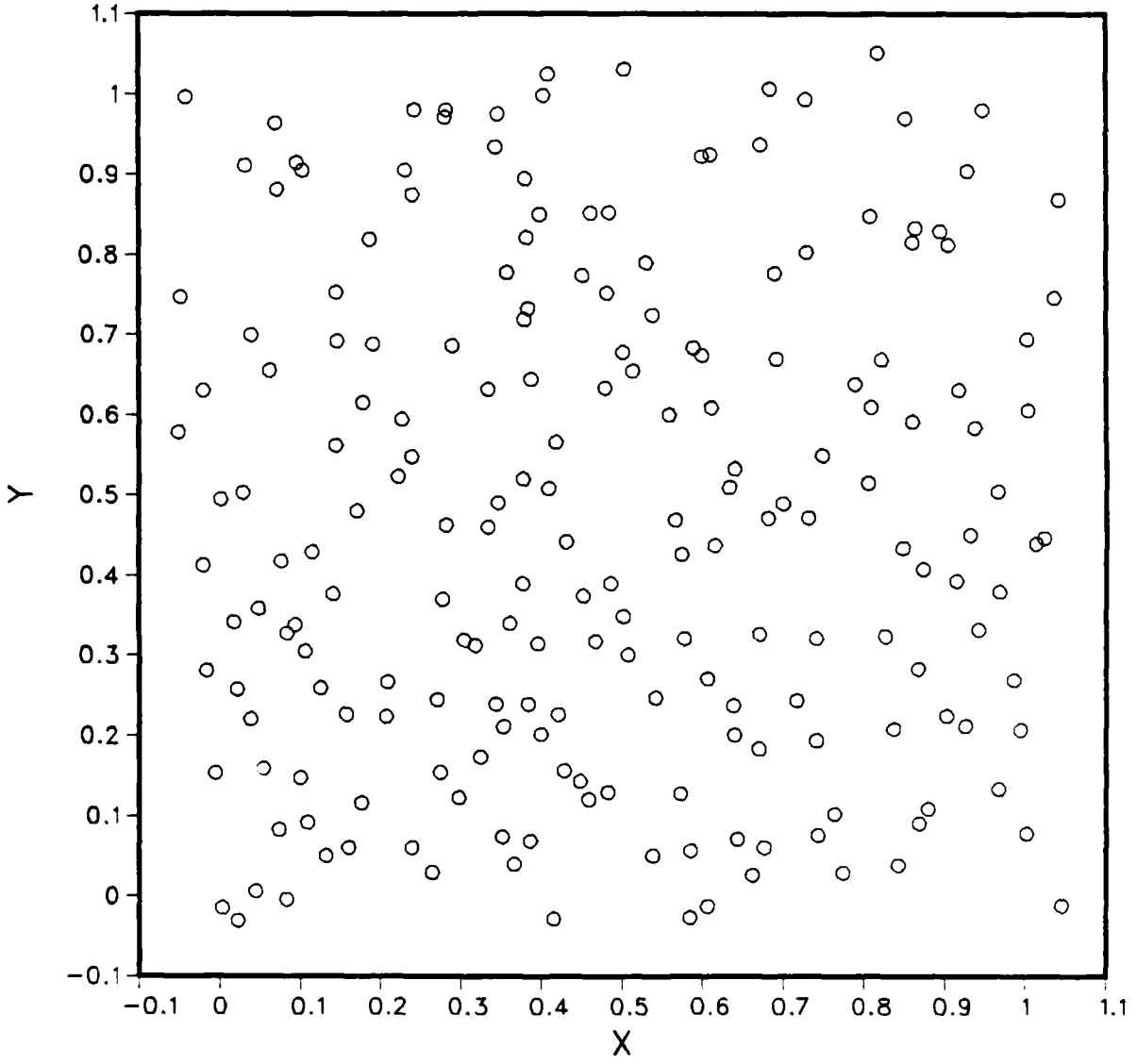


Figure 5.5. 200 Point Humps & Dips Data Set.

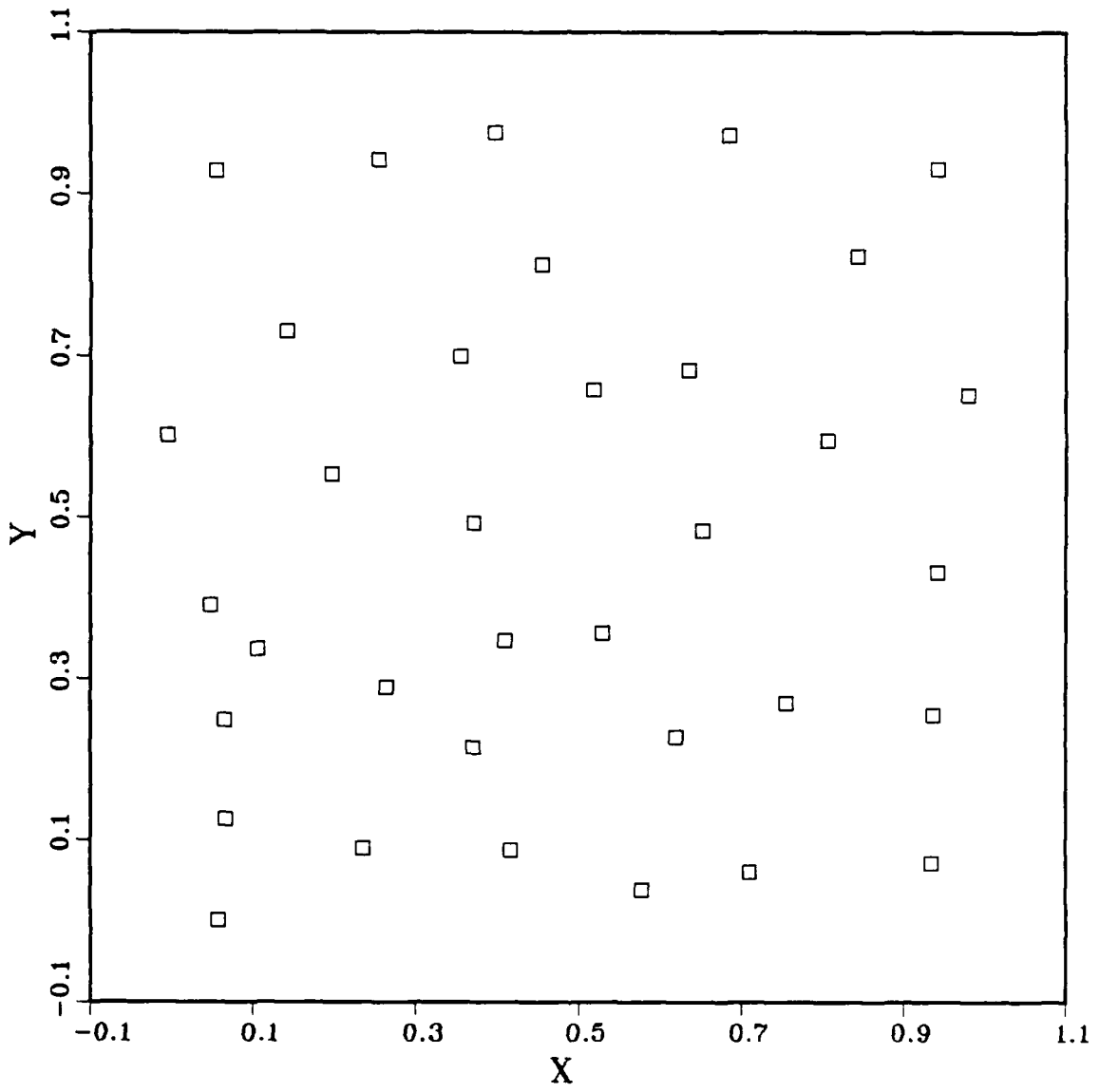


Figure 5.6. 35 Knots Representing 200 Point Humps & Dips Set.

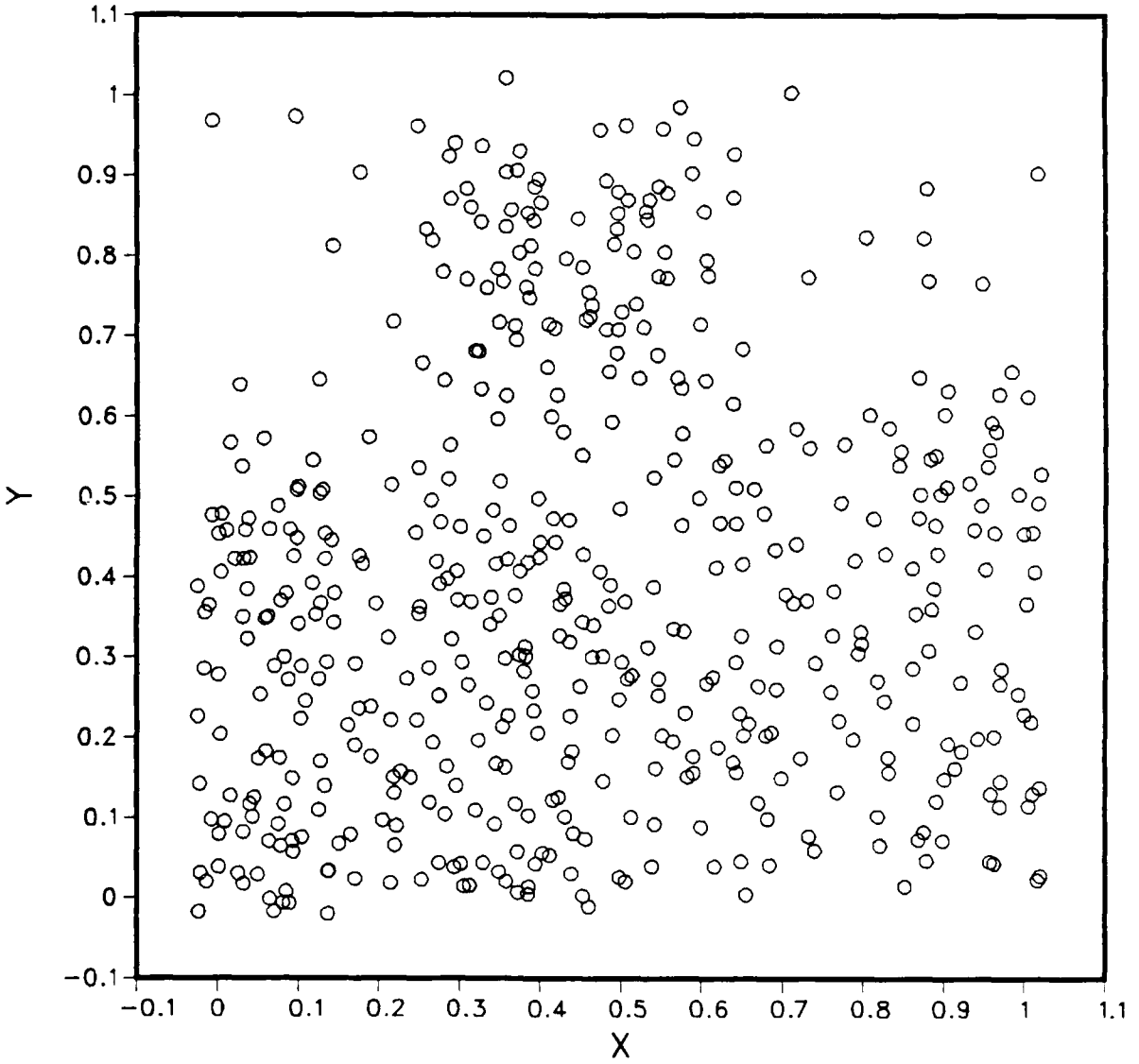


Figure 5.7. 500 Point Humps & Dips Data Set.

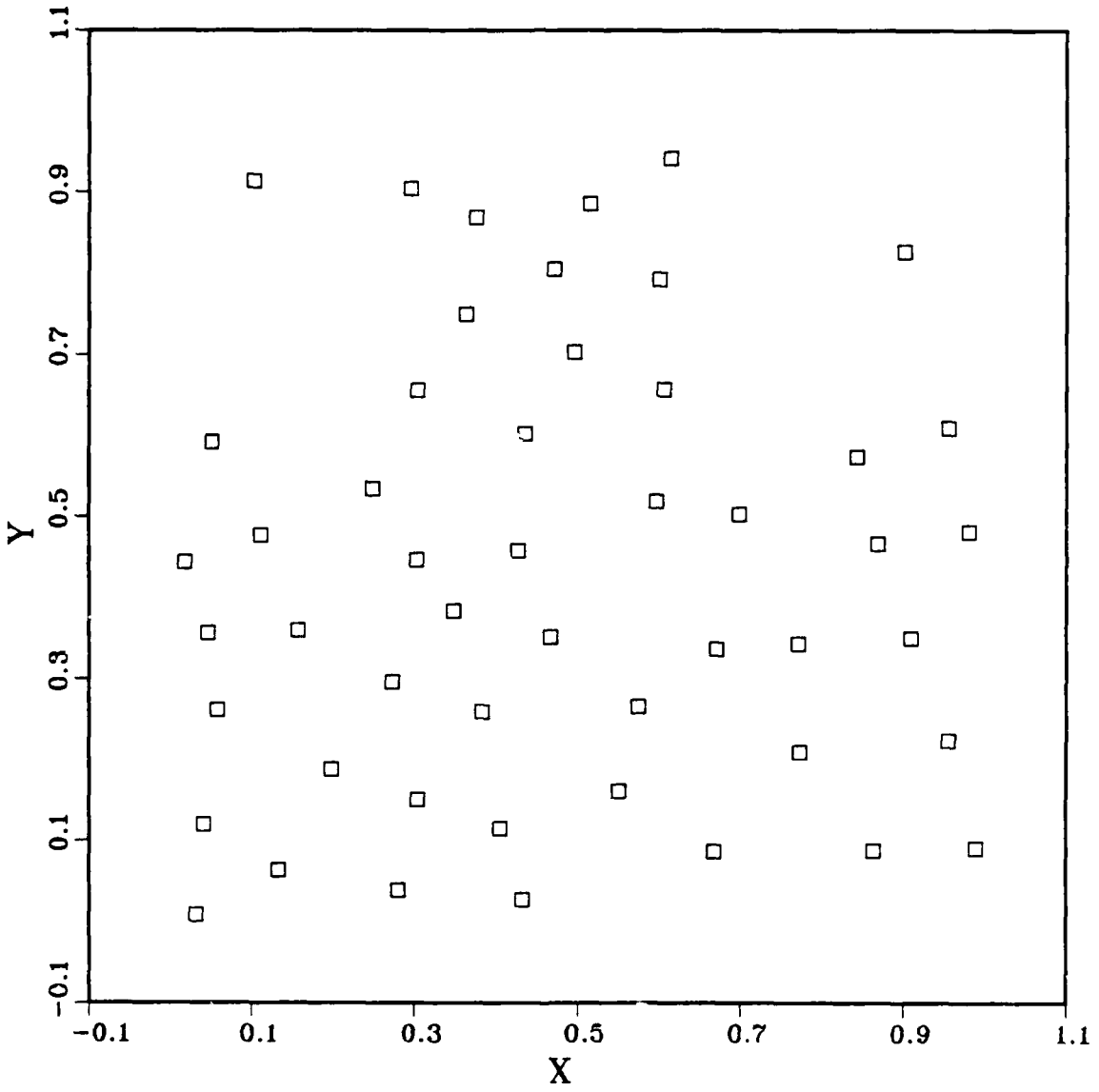


Figure 5.8. 50 Knots Representing 500 Point Humps & Dips Set.

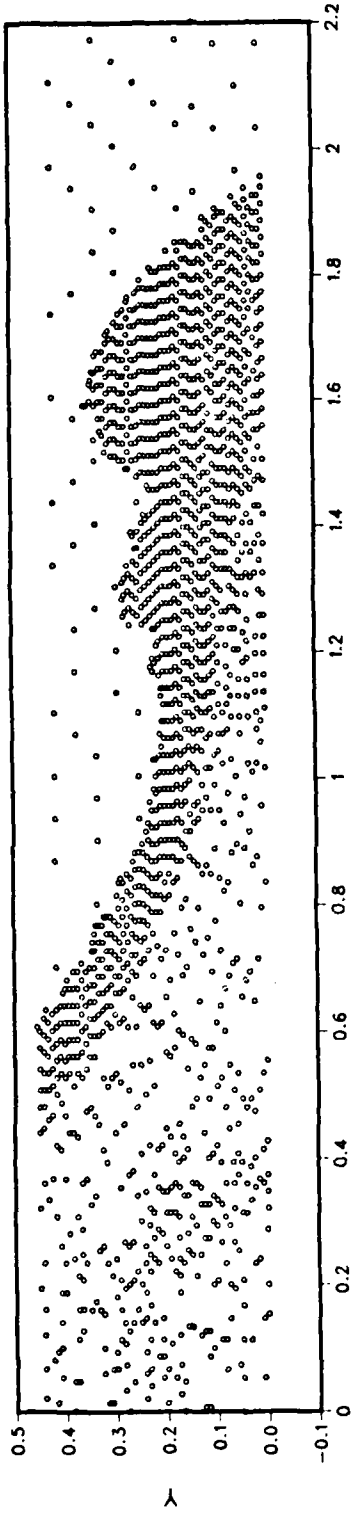


Figure 5.9 1669 Point Hydrographic Data Set

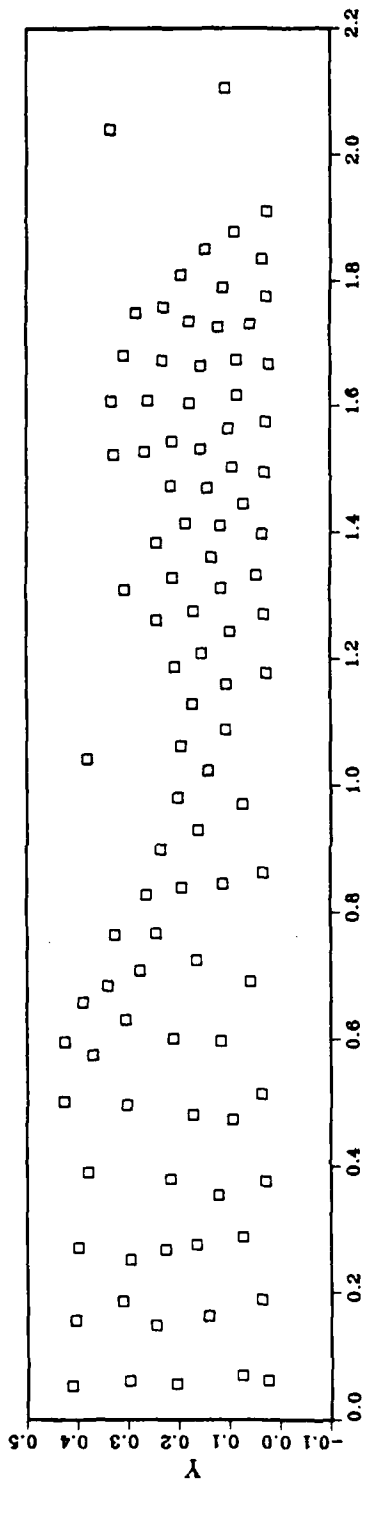


Figure 5.10 100 Knots Representing 1669 Point Hydrographic Set

above to create several 200 point sets of data on which to experiment. The 500 point set was generated using the density proportional to curvature method as well.

Additionally, we present a series of even numbered figures, (5.2 through 5.10), which show various final knot locations for the previously mentioned data sets. By comparing these figures to the corresponding parent sets of data, we can obtain a 'feel' for how reasonable the representative sets actually turn out to be using the knot selection process outlined here. The numbers of knot points used in representing these data sets were selected so as to achieve different combinations for comparison and discussion; they are not meant to intimate that a certain number of knot points is optimal in representing a specific set of data.

C. APPROXIMATING THE SURFACE

In discussing the approximating surface, we are interested in knowing how close the constructed surface will fit the underlying surface. First, we consider what we should expect to find knowing a little about least squares approximation and the RMS error. Each of the data sets considered thus far were used to generate the third component at each point, the dependent variable. These can be made 'exact' using a known function or 'contaminated,' using independent, normally distributed random errors in the dependent variable. In all the data sets presented above, the dependent variables of the data sets were subjected to random errors using a composite standard deviation of less than 0.05. This lead to two versions of the same sets of data: one without errors, and the other 'contaminated' with small 'measurement' errors. We wished to experiment by comparing runs of the

driver program using these data sets to verify what one might expect to observe in such experiments as described below.

For each of the experiments, the maximum, mean and RMS errors were computed on the residuals, as well as on a 33 by 33 square grid for comparison purposes, although we present only the RMS error results. In general, we would expect to see an overall decrease in the RMS error on both the residuals and the grid as the number of knot points used to represent the data is increased. Assuming we have data contaminated with errors, we know that each data point is the sum of an unknown underlying function value and an error function value. Thus, when we approximate the surface using a least squares method, we anticipate that the difference (closeness of fit) between the constructed surface at the data points and the 'true' surface defined by the unknown underlying function is entirely due to the presence of error in the data, and not attributable to any 'leeway' in the constructed surface. If this were the case, then the RMS error in the residuals would match the composite standard deviation injected into the contrived contaminated data. Furthermore, since we are also analyzing the closeness of fit of the constructed function to the unknown underlying function at the grid points, we would also expect the RMS error on the grid to be smaller than the same composite standard deviation, since the grid sample is larger and the errors are distributed more evenly throughout the entire region of interest.

In the idealistic case where no errors occur in the data, we know that the sort of behavior described above cannot be manifested and that the difference between the constructed surface at the data points and the 'true' surface is entirely due to 'slack' in the constructed surface. Here, we

anticipate that the RMS error in the residuals is approximately equal to the RMS error at the grid, thereby giving evidence that the error in the constructed surface is uniformly distributed over the entire region of interest, a desirable characteristic.

We have included the results obtained using the smoothing spline and another method known as the BiCubic Hermite Approximation method (BHASHD), for both of the 200 point data sets, and the BHASHD method on the 500 point data set. [Ref. 18] These runs were made on data sets whose dependent variables were subjected to errors, and without errors, in order to compare the results of the least squares method using varying numbers of knot points to several 'outside' control methods. The smoothing spline technique used in obtaining these results was described earlier in Chapter 2 and is detailed in Wendelberger. [Ref. 19]

The BHASHD method involves local least squares polynomial fits to estimate the value of the function, the two first partial derivatives, and the cross partial derivatives (as a measure of the 'twist' in the surface), on a grid of input points. In the usual mode, a second degree polynomial is used at the interior grid points, and a first degree polynomial is used at the 'boundary' grid points. For a 5 x 5 input grid, for example, there would be 100 parameters using this method: 25 points at which an approximate value of the surface is made; 25 points at which the gradients in the x and y directions are computed; and 25 points at which the twist in the surface is computed. A bicubic Hermite Approximation on this grid is then used as the approximating function. The 5 x 5 input grid replaces the optimization of the knot point locations done in least squares TPS, but as

with the least squares method, a 33 by 33 output grid of values is computed for the surface at which a RMS error value is calculated. We would not expect to see any evidence of smoothing in the BHASHD method since the method is local in nature and does not take the entire set of data into account as it constructs the approximating surface.

Tables III through V summarize the results attained, and in light of the foregoing discussion, we make the following observations. When appropriate, in place of the number of knot points used, we have used the size of the input grid for the BHASHD method, and the acronym Smthng for the smoothing spline method.

1) The general trend of the RMS on both the residuals and the grid is to decrease as the number of knot points is increased. As expected with the data without errors, the RMS of the residuals and the RMS on the grid are roughly equivalent. Thus, we can conclude that the error in the constructed surface is nearly uniformly distributed over the entire region of interest.

2) In the data contaminated with errors, we again see what was predicted: the RMS of the residuals matches the composite standard deviation of the data points, and the RMS on the grid is somewhat smaller than the RMS of the residuals, although we cannot attribute the entire difference to the injected error in the data. Here, we witness a phenomenon called undersmoothing, wherein the constructed surface tends to fit the error rather than the data. We note an anomaly in the results for the contaminated 200 point cliff data set where the RMS error on the grid increases as the number of knot points increases.

3) We see that for the case where the data is exact, the smoothing spline method yields a residual RMS value of 0.0, which could be expected, since there is no error in the data. On the grid, the RMS is small in both of the exact data examples, since some small amount of error on the grid is expected. However, in the case where the data is contaminated, we observe that not until we use a 'large' number of knots do we begin to approach the results attained in the smoothing spline method through the use of the least squares method. The fact that we can begin to approach the smoothing spline results using the least squares spline is an accomplishment in and of itself. But we do need many more knot points than was originally indicated in the square root of the number of data points criterion mentioned in Chapter 4. We

TABLE III

COMPARISON OF RMS ERRORS ON 'HUMPS & DIPS' 200 PTS

Number of Data Pts/ Knot Pts	No Errors in Data		Contaminated Data	
	Residual	Grid	Residual	Grid
200/20	.05525	.05465	.07571	.05866
200/25	.02520	.02646	.05603	.03385
200/5 x 5	.01206	.01332	.04819	.04965
200/35	.01662	.01843	.05274	.02853
200/6 x 6	.00968	.01144	.05028	.03962
200/Smthng	0.0	.00254	.03900	.02789

TABLE IV

COMPARISON OF RMS ERRORS ON 'CLIFF' 200 PTS

Number of Data Pts/ Knot Pts	No Errors in Data		Contaminated Data	
	Residual	Grid	Residual	Grid
200/20	.01562	.01474	.05214	.01795
200/25	.01179	.01154	.04805	.02040
200/5 x 5	.00777	.00613	.05996	.04819
200/35	.00626	.00616	.04590	.02146
200/6 x 6	.00512	.00417	.05113	.03745
200/Smthng	0.0	.00096	.04272	.01806

TABLE V

COMPARISON OF RMS ERRORS ON 'HUMPS & DIPS' 500 PTS

Number of Data Pts/ Knot Pts	No Errors in Data		Contaminated Data	
	Residual	Grid	Residual	Grid
500/20	.02402	.02517	.05256	.02738
500/25	.01664	.01766	.04818	.02283
500/5 x 5	.01346	.01230	.05844	.03767
500/50	.00645	.00845	.04544	.01961
500/7 x 7	.00645	.00552	.05696	.04864

also remark that as before, the smoothing spline cannot be used for more that about 200 data points due to the fact a large system of equations must be solved.

4) For the data without errors, we see that the RMS error on both the residuals and the grid are approximately half those of the least squares method using a nearly equal number of knot points. For the contaminated data, the RMS error on the residuals in the BHASHD method is nearly equal to the composite standard deviation injected into the data. However, on the grid, the least squares method does better, an indication that smoothing is occurring in the least squares case, while little is occurring in the BHASHD method, as anticipated. We also note that an increase in the number of input grid points does not seem to significantly improve the RMS errors in the BHASHD method, even though an increase in the number of knots in the least squares method usually yields improved results.

D. TIMING

The amount of time a particular program run requires to find the 'best' knot configuration depends on the number of data points and the search pattern followed, which in turn is a function of the initial guess for the knots. Furthermore, there does not seem to be any consistent correlation between the number of knot points used for a given data set and the time taken to complete the search as seen in Table VI. The 100 point set, for

example, required 12 seconds using 10 knots, 38 seconds using 20 knots, only 13.5 seconds using 25 knots, and more than 468 seconds using 50 knots. Obviously, there does appear to be a 'point' of diminishing returns at which the amount of time necessary to conduct a full search in accordance with the TWEEK algorithm becomes unfeasible.

We tested the program on a data set containing 1669 points derived from a larger set of hydrographic data collected in and about Monterey Bay using both 50 and 100 knot points. The second run required more than the allotted amount of CPU time and needed to be run as a batch job. This lead to several slight modifications of the driver program, including a provision for outputting the 'best set of knots to date' at intervals of 13 plus minutes (chosen as a reasonable initial deadline time), anticipating the event wherein the searching pattern required a large, unpredictable amount of computer time. Since we wished to have the program continue with its current searching pattern, we specified that the output occur at a convenient time in the code; we chose the point immediately before a new determination of the knots with the most and fewest data points at the top of an iteration within the TWEEK subroutine. The driver program makes available the option of inputting this 'best set of knots to date' to accomodate the continuation of the search at the place where it had prematurely ended.

The time required to set up the coefficient matrix, perform the QR decompostion, and obtain the least squares solution via the LSCOEF subroutine is given in the column labeled 'COMP1'. These times are generally

TABLE VI

TIMING COMPARISONS

Data Set Descptn	Number of Knot Pts	Search Time (secs)	Comp 1 (secs)	Comp 2 (secs)
100 (Original Random Surface)	10	12.393	.069	.146
	20	38.185	.133	.339
	25	13.724	.169	.415
	50	468.119	.525	.821
200 (Humps & Dips)	20	28.896	.246	.316
	25	39.992	.312	.399
	35	73.092	.479	.579
200 (Cliff)	20	27.432	.246	.319
	25	37.506	.339	.415
	35	253.996	.505	.602
500 (Humps & Dips)	25	50.672	.708	.405
	50	99.909	1.827	.808
1669 (Mty Bay)	50	828.86		
	100	3438.69		

in consonance with the size of the coefficient matrix involved, which is N+3 by K+3. The time required to evaluate the newly constructed function F, given the coefficient matrix, on a rectangular grid of points via the GEVGRD subroutine is given in the column labeled 'COMP2'. As expected, these times are also consistent with the task at hand.

When the 1669 point set of hydrographic data was run, we encountered a situation wherein a steep gradient between adjacent data points occurred; this happens near one of the corners of the region of interest, which

extends over the Monterey undersea canyon. We note that the data set was collected using regular intervals along sounding lines in accordance with normal hydrographic data collection procedures. This lead to a relatively large maximum error over the residuals even though the RMS error over the residuals remained relatively small, an obvious discrepancy for what should have been a good fitting constructed surface, which would have yielded a consistently small set of error values. This phenomenon is observed because a large change in the dependent variable occurred in an area where the data was more or less uniform, thus violating our basic underlying assumption. By this assumption, we should have had many more data points in this area since 'something very interesting is happening to the dependent variable.' More importantly, as we have mentioned, the density of the knot points is seen to be (Figure 5.10) correspondingly uniform which means that the number of basis functions used in that area would be no different than in the rest of the region. Thus, in an area where the number of basis functions needed to construct a good fitting surface has increased, we have maintained the same number. The result is an inaccurate, poorly constructed surface which can have large amounts of slack where little slack should be allowed.

E. CONCLUSIONS

We have noted the 'running' time required for the various experimental data sets, and it is obvious that the efficiency of the algorithm is lacking. It is fair to deduce that the algorithm is probably conducting some of the searching effort in areas which need not be searched, and it may be that the algorithm is not searching in some areas of greater potential.

In terms of where to conduct a better search, there is currently not a good answer. However, in terms of how to proceed in order to conduct a

better search, some suggestions for future research efforts will be given. Since we are currently checking all of the possible combinations of high and low density knot points, the inefficiency is most likely occurring in the form of some unnecessary checks. Hence, it would be prudent to design an algorithm which followed the same general scheme with the following exception. Instead of checking all possible combinations, first move the knot with the most data points toward the nearest knot with the fewest data points, and vice-versa, in the same sort of symmetric manner as before. Ultimately, the process needs to be pared down again, and this approach presently holds the most promise.

Finally, as was noted in Chapter 3, there is the matter concerning the tie breaking criterion, which is by default in this algorithm. A more reasonable approach may be to divide those data points which reside on the boundary of the final Dirichlet Tesselation among the knot points having a 'legitimate' claim. Taking this approach would alleviate the arbitrariness of the current method and possibly lead to an even smaller GN^2 function value. Alternatively, this deficiency could be addressed as it was in the one dimensional example, wherein the alternative data point assignment is actually checked. However, the magnitude of this deficiency is not overwhelming, in that it only requires attention when the final Dirichlet Tesselation contains one or more data points on tile boundaries.

LIST OF REFERENCES

1. Davis, P. J., Interpolation and Approximation, Dover Publications Inc., 1975.

2. Schumaker, L. L., "Fitting Surfaces to Scattered Data," Approximation Theory II, eds. G. G. Lorentz, C. K. Chui, L. L. Schumaker, Academic Press, 1976.

3. Franke, R., "Scattered Data Interpolation: Tests of Some Methods," Mathematics of Computation, V. 38, No. 157, January 1982.

4. Gerald, C. F. and Wheatley, P. O., Applied Numerical Analysis, 3d ed., Addison-Wesley Publishing Co., 1984.

5. Giordano, F. R. and Weir, M. D., A First Course in Mathematical Modeling, Brooks/Cole Publishing Co., 1985.

6. Strang, G., Linear Algebra and its Applications, 2d ed., Academic Press Inc., 1980.

7. Ahlberg, J. H., Nilson, E. N., and Walsh, J. L., The Theory of Splines and Their Applications, Academic Press Inc., 1967.

8. de Boor, C., A Practical Guide to Splines, Applied Mathematical Sciences V. 27, Springer-Verlag New York Inc., 1978.

9. Wahba, G. and Wendelberger, J., "Some New Mathematical Methods for Variational Objective Analysis Using Splines and Cross Validation," Monthly Weather Review, V. 108, August, 1980.

10. Harder, R. L. and Desmarais, R. N., "Interpolation Using Surface Splines," J. Aircraft, V. 9, No. 2, February, 1972.

11. Duchon, J., "Splines Minimizing Rotation Invariant Semi-Norms in Sobelov Spaces," Constructive Theory of Functions of Several Variables, eds., W. Schemepp and K. Zeller, Lecture Notes in Math 571, Springer, 1977.

12. Meinguet, J., "Surface Spline Interpolation: Basic Theory and Computational Aspects," Approximation Theory and Spline Functions, eds. S. P. Singh, et al., D. Reidel Publishing Co. 1984.

13. Epstein, B., Linear Functional Analysis, W. B. Saunders Co., 1970.

14. Mansfield, L. E., "On the Optimal Approximation of Linear Functionals in Spaces of Bivariate Functions," SIAM J. Numer. Anal., V. 8, No. 1, March, 1971.

15. Lawson, C. L. and Hanson, R. J., Solving Least Squares Problems, Prentice-Hall, Inc., 1974.

16. Dongara, J. and others, LINPACK User's Guide, SIAM, 1979.

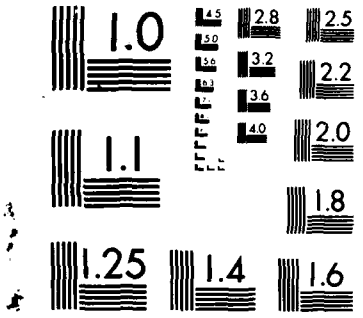
17. Golub, G. H. and Van Loan, C. F., Matrix Computations, The Johns Hopkins University Press, 1983.

18. Foley, T. A., "Scattered Data Interpolation and Approximation," Lawrence Livermore National Laboratory Rpt. No. UCID-20346, February 11, 1985.

19. Wendelberger, J., The Computation of Laplacian Smoothing Splines with Examples, TR No. 648, Dept. of Statistics, Univ. of Wisconsin, Madison, September, 1981.

AD-A174 507 KNOT SELECTION FOR LEAST SQUARES APPROXIMATION USING THIN PLATE SPLINES(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA J R MCMAHON JUN 86 2/2

UNCLASSIFIED F/G 12/1 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5000	2
3.	Professor Richard Franke, Code 53Fe Department of Mathematics Naval Postgraduate School Monterey, California 93943-5000	5
4.	Professor Charles Schelin, Code 53Sh Department of Mathematics Naval Postgraduate School Monterey, California 93943-5000	1
5.	Professor Gordon Latta, Code 53Lz Department of Mathematics Naval Postgraduate School Monterey, California 93943-5000	1
6.	Professor G.M. Nielson Department of Computer Science Arizona State University Tempe, Arizona 85287	1
7.	Professor Tom Foley Department of Computer Science Arizona State University Tempe, Arizona 85287	1
8.	Professor Grace Wahba Department of Statistics University of Wisconsin Madison, Wisconsin 53705	1
9.	Professor Florencio Utreras Depto. Matematicas Y Ciencias de la Computacion Casilla 170/3-Correro 3-Santiago Chile	1

10. Dr. Licia Lenarduzzi 1
IAMI
Via Cicognara 7
20129 Milano
Italia

11. Colonel David Cameron 2
Head, Department of Mathematics
United States Military Academy
West Point, New York 10966

12. Director 1
US Army Engineer Topographic Laboratory
Fort Belvoir, Virginia 22060

13. Captain John R. McMahon 10
31 Commodore Lane
East Falmouth, Massachusetts 02536

END

12-86

DTIC